

Možnosti 3D stereo projekce v Blenderu

The Possibility of 3D Stereo Projection in Blender

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Tomáš Popek**
Studijní program: N2647 Informační a komunikační technologie
Studijní obor: 2612T025 Informatika a výpočetní technika
Téma: **Možnosti 3D stereo projekce v Blenderu**
The Possibility of 3D Stereo Projection in Blender

Zásady pro vypracování:

S příchodem nových technologií umožňující 3D stereo projekci, vznikají také rostoucí požadavky na programy, jejich funkčnost a nabízené možnosti. Cílem této práce je zaměřit se na možnosti 3D stereo projekce profesionálního open source nástroje Blender a jeho možnosti zobrazování.

1. Nastudujte problematiku 3D stereo projekce a možnosti stereo zobrazování profesionálního open source nástroje Blender (panel 3D view, game engine, stereo animace apod.).
2. Upravte stávající verzi Blenderu (zdrojové kódy, C++, OpenGL) tak, aby umožňovala zobrazení hlavní 3D pracovní plochy, používané při návrhu a modelování scény, pomocí 3D stereo projekce. Pro ověření využijte 3D stereo projektor v laboratořích pro počítačovou grafiku.
3. Otestujte použití rozšíření stereo projekce do Blenderu s využitím možností Add-ons (Python).
4. Přínos stereo zobrazení ověřte na vzorku uživatelů programu Blender a výsledky shrňte ve své práci.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Martin Němec, Ph.D.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 2012

.....
.....

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla. Obzvláště pak mému vedoucímu panu Ing. Martinu Němcovi, Ph.D. a panu prof. Ing. Václavu Skálovi, CSc. za jejich podporu a rady.

Abstrakt

Cílem této diplomové práce je vytvořit a otestovat modifikaci open source nástroje Blender, která zajistí podporu základních typů stereoskopické projekce. Jsou zde rozepsány využívané stereoskopické metody a jejich praktická implementace v aplikaci. Po úpravě je možné využít stereoskopické zobrazení při tvorbě a modelování přímo ve vykreslované scéně. I když Game Engine, jakožto jedna z částí Blenderu, již podporu stereoprojekce obsahuje, je určena jen jako možnost zobrazení výsledných aplikací a neumožňuje využít stereo zobrazení při návrhu a tvorbě 3D modelů, scén nebo animací. Za přidáním stereoprojekce k funkcím programu Blender stojí snaha rozšířit uživatelům stávající možnosti při modelování. V závěru dokumentu jsou popsány jednotlivé praktické testy nově vytvořeného přídatku. Zároveň jsou zde zmíněny některé problémy, na které se v průběhu práce narazilo a bylo potřeba je vyřešit.

Klíčová slova: Anaglyph, Blender, OpenGL, Stereoprojekce, Stereoskopie, Quad buffer

Abstract

The aim of this thesis is to develop and test a modification of Blender, an open source tool, to ensure support for the primary types of stereoscopic projection. There are described used stereoscopic methods and their practical implementation in the application. After adjustment, the stereoscopic scheme can be used for production and modeling directly in the scene. Even though the game engine, an integrated part of Blender, already contains stereoscopic projection support, it is only designed as a way to view complete applications and does not facilitate the use of the stereoscopic scheme in the design of three-dimensional models, scenes or animations. Behind adding of stereoprojection to Blenders features is an effort to expand the existing users possibilities in modeling. At the end of the document is described each practical test of newly developed extension. Also, there are discussed some of the problems, that we encountered in this work, and their solutions.

Keywords: Anaglyph, Blender, OpenGL, Stereoprojection, Stereoscopy, Quad buffer

Seznam použitých zkratk a symbolů

GHOST	– Generic Handy Operating System Toolkit
GLUT	– OpenGL Utility Toolkit
NaN	– Not A Number
NURBS	– Non-Uniform Rational Basis Spline
OpenGL	– Open Graphics Library
WSEAS	– World Scientific and Engineering Academy and Society

Obsah

1	Úvod	5
2	Aktuální stav	6
2.1	GHOST	7
2.2	Game Engine	7
2.3	Blender player	8
3	Prostorové vidění	9
3.1	Stereoslepota	11
3.2	Zásady pro vytváření stereoskopických snímků	11
3.3	Paralaxa	12
3.3.1	Nulová paralaxa	12
3.3.2	Pozitivní paralaxa	13
3.3.3	Negativní paralaxa	14
3.3.4	Divergentní paralaxa	15
3.4	Metody získávání stereopárů	15
3.4.1	Toe-in	15
3.4.2	Parallel	16
3.4.3	Off-axis	17
4	Stereoskopická projekce	18
4.1	Pasivní projekce	18
4.1.1	Anaglyph	18
4.1.2	Polarizace	19
4.2	Aktivní projekce	19
4.3	Autostereoskopický displej	19
5	Implementace	21
5.1	Creator	22
5.2	Context	23
5.3	Window manager	24
5.3.1	Stereokurzor	25
5.3.2	Implementace kurzoru	26
5.4	Nastavení kamery	27
5.4.1	Využitá metoda	28
5.4.2	Paralaxa	28
5.5	Viewport	30
5.5.1	Anaglyph	31
5.5.2	Vertical interlace	33
5.5.3	Horizontal interlace	34
5.5.4	Quad buffer	35
5.6	Úprava klávesových zkratk	37

6	Python	39
6.1	Skript	40
7	Testování	41
7.1	Prvotní testy	41
7.1.1	Anaglyph	42
7.1.2	Quad buffer	43
7.2	Testy po úpravách	44
7.3	Hardware	46
8	Závěr	48
9	Reference	50
	Přílohy	52
A	UML struktura	53
B	Dotazník	54
C	Výsledky dotazníku	55

Seznam obrázků

1	Původní program Blender	6
2	Překrývání, lineární perspektiva, relativní pohyb[23]	10
3	Binokulární disparita[23]	10
4	Nulová paralaxa	13
5	Pozitivní paralaxa	13
6	Negativní paralaxa	14
7	Divergentní paralaxa	15
8	Toe-in metoda	16
9	Paralelní metoda	16
10	Off-axis metoda	17
11	Je důležité určit, kdy je vhodná pozitivní a kdy negativní paralaxa[9]	29
12	Získání stereopáru paralelní metodou	29
13	Tvorba stereoprojekce metodou anaglyph	32
14	Tvorba stereoprojekce prokládanou metodou	35
15	Tvorba stereoprojekce metodou quad buffer	37
16	Uživatelské rozhraní	40
17	Rozdíly mezi wireframe (vlevo) a solid (vpravo) zobrazením	42
18	Výsledky testů metody anaglyph	43
19	Výsledky první vlny testů metody quad buffer	44
20	Výsledky testů po úpravě metody quad buffer s využitím projektoru	45
21	Výsledky testů po úpravě metody quad buffer s využitím monitoru	45

Seznam výpisů zdrojového kódu

1	Registrace a zpracování parametrů	22
2	Práce s proměnnými	24
3	Úprava transformační matice	28
4	Nastavení vertikálních prokládacích masek	34
5	Nastavení horizontální prokládací masky	35
6	Registrace klávesových zkratk a definice jejich funkcí	38

1 Úvod

S rozvojem informačních technologií rostou také možnosti využívání 3D technologií a jejich zobrazování. Panuje určitá tendence o to, přiblížit uživatelům promítané prvky realističtěji a plastičtěji. Dalo by se předpokládat, že se v této oblasti jedná o přirozený vývoj. Kdysi se vizuální informace přenášely kresbami, později se objevily první fotografie, a následně i němé černobílé filmy. S postupným vývojem technologií byl brzy pro lepší prožitek do obrazu přidán zvuk a později i barva. Je tedy přirozené, když se dnešní snaha o úpravy těchto projekcí snaží orientovat i na přidání dalších možností a rozměrů. S moderním 3D systémem pro stereoskopickou projekci jsme se mohli setkat například v kinech při sledování některých filmů za pomoci speciálních brýlí. Dnes je již možné využít takový způsob zobrazování i při práci na počítačích, které tuto technologii podporují. Existují také televizory, které podporují stereo vysílání některých filmů nebo pořadů. Stereoskopie je technika, která se v poslední době rozmáhá a snaží se konkurovat klasickému sledování běžných filmů, prezentací nebo hraní her.

Tato práce by se dala rozdělit na tři samostatné oddíly. První část se zabývá převážně teorií a snaží se nastínit řešené problémy. Ve druhé části se řeší praktická implementace programu. Poslední část této práce je zaměřena na testy, které byly na výsledné aplikaci prováděny.

V kapitole zabývající se aktuálním stavem programu Blender bude popsána jeho historie a stávající možnosti. Zároveň zde budou nastíněny jednotlivé části programu, které jsou určitým způsobem důležité pro pochopení dané problematiky. Po sekci věnované teorii prostorového vnímání následuje popis obecné problematiky stereoskopie a jejích možností při praktické implementaci. V této části budou také popsány různé metody získávání stereoskopických obrazů, a případně i jejich výhody a nevýhody. V závěru této kapitoly se nachází popisy některých důležitých druhů stereoprojekce jako takové.

V následující sekci věnované implementaci bude podrobně rozepsáno zmapování důležitých částí programu Blender, které byly při tvorbě využity a upraveny. Budou popsány jednotlivé kroky, které byly provedeny za účelem získání funkční stereoskopické verze tohoto programu. Mezi tyto kroky bude patřit například volba způsobu získávání stereoskopických snímků, úprava kamery snímající danou scénu nebo zpracování jednotlivých projekčních metod. Dále bude popsána tvorba ovládacích prvků a uživatelského rozhraní.

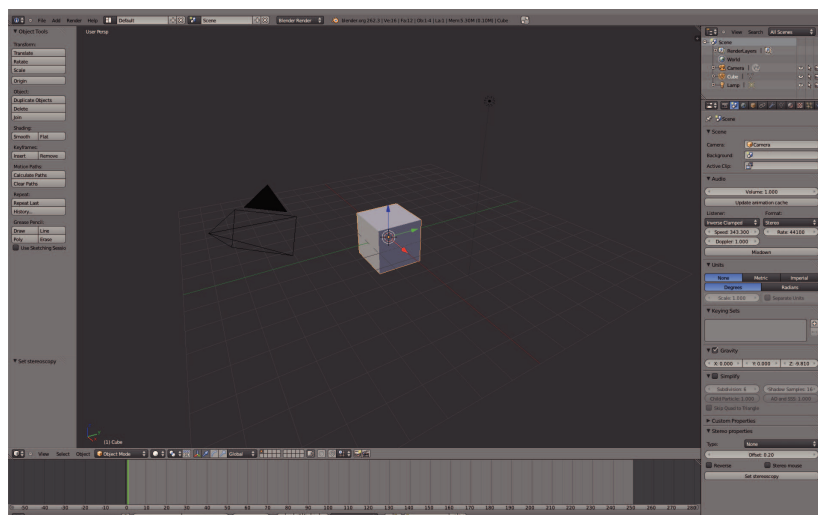
Před závěrem této práce se nachází část popisující prováděné testy výsledného programu. V této části budou rozepsány problémy, na které se v průběhu těchto testů narazilo a způsoby jejich řešení. Rovněž zde bude popsáno, jaký hardware byl pro tyto testy použit. Závěrem je shrnut výsledek celé diplomové práce. Jsou zde popsány možnosti použití a nápady možných budoucích kroků v tomto oboru. V poslední části dokumentu jsou k nahlédnutí využívané reference, které hlouběji rozebírají celkovou problematiku nastíněnou v této práci. V přílohách umístěných na samém konci tohoto dokumentu je přiložena finální verze dotazníku, který byl využíván při praktických testech, a také tabulky se zpracovanými výsledky těchto dotazníků. Mezi přílohy je také umístěn diagram znázorňující provázanost jednotlivých částí programu, které jsou pro tuto práci důležité.

2 Aktuální stav

Program Blender je zdarma dostupný open source software pro 3D modelování, tvorbu animací a renderování scén nebo modelů. Pro zobrazování svého interface Blender využívá knihovnu OpenGL. Díky použití OpenGL a dalších modulů je Blender silným a robustním nástrojem nezávislým na platformě nebo operačním systému[6].

Blender byl vyvíjen jako vlastní aplikace holandského animačního studia NeoGeo a NaN. Program byl původně distribuován jako shareware, společnost NaN však v roce 2002 zkrachovala. Později bylo odsouhlaseno vypuštění Blenderu pod hlavičkou GNU General Public License a v polovině roku 2002 byl zdrojový kód Blenderu uvolněn. Blender je v současné době aktivně vyvíjen pod dohledem Blender Foundation.

Modelovací schopnosti programu jsou zaměřeny především na práci s ploškovou reprezentací těles. Blender umožňuje pracovat s takzvanými subsurf plochami, které jsou uživateli s oblibou využívány. Dále pak podporuje v omezenější formě práci s parametrickými plochami a křivkami (Bezier, NURBS) nebo implicitními plochami[8].



Obrázek 1: Původní program Blender

Experimentálně již existuje upravená verze staršího vydání Blenderu 2.45, kterou vytvořila Západočeská univerzita v Plzni. Tato práce spočívala ve tvorbě nastavby nad klasickým programem Blender. Jejich cílem bylo vytvořit projekci pracovní plochy tak, aby bylo možné ji zobrazovat ve 3D s využitím stereo notebooku SHARP. Samotného stereoskopického zobrazení bylo docíleno s využitím prokládaného řádkování. Využívá se zde úpravy zobrazovacího bufferu, který promítá vykreslovaný obraz na monitor. Ve výsledku upravili nastavení tohoto bufferu tak, že byli schopni získat dva odsazené obrazy, které se pak prolnouly s využitím GLSL shaderu. Toto prolnutí pak promítali jako výsledný obraz viewportu na monitor. S příchodem nové verze Blenderu byla již tato

metoda nepoužitelná, neboť s novou verzí došlo také k přepracování samotného zdrojového kódu Blenderu[26].

2.1 GHOST

Mimo OpenGL Blender využívá vlastní nástroj s názvem GHOST. Tato knihovna byla vytvořena, aby zastoupila původní OpenGL knihovnu GLUT, kterou Blender využíval do dob, než byla potřeba rozšíření funkčnosti i na operační systémy Mac OSX firmy Apple. Z důvodu převodu na jiný operační systém byla vytvořena knihovna GHOST, která plnohodnotně zastoupila GLUT ve všech aspektech, které Blender potřeboval.

Důvod, proč je tato knihovna zmiňována, je ten, že právě na ní, a na práci s ní, byla postavena programovací část této diplomové práce. Jelikož je GHOST plnohodnotnou náhradou knihovny GLUT, dokáže zpracovávat všechny potřeby programu Blender:

- Management časovačů.
- Management oken.
- Management událostí.
- Management tvarů a stavů kurzoru.
- Přístup k tlačítkům myši, kolečku a informacím o klávesnici.

Samotný GHOST je psán v programovacím jazyce C++ s tím, že je jeho API zapouzdřeno v klasickém C[25].

2.2 Game Engine

Blender obsahuje samostatnou část s názvem Game Engine, uvnitř které může uživatel vytvářet různé interaktivní aplikace jako hry, virtuální prohlídky scén a podobně. Game Engine má v sobě od verze 2.40 integrovaný fyzikální engine Bullet, s jehož pomocí lze věrně simulovat dynamiku pevných těles. Zároveň je zde implementovaná podpora skriptů v jazyce Python, GLSL shaderů nebo zvuků s využitím knihovny OpenAL. Stejně jako v Blenderu samotném je i zde pro vykreslování využita knihovna OpenGL[8]. Jedny z posledních verzí tohoto engine již obsahují zabudovanou možnost jednoduché stereo-projekce. Tato stereoprojekce se dá využít pouze při zapnutí „game módu“ a má několik typů zobrazování:

- **Vinterlace** – Vertikální prokládání řádků.
- **Side-by-side** – Metoda se zobrazením obrazů vedle sebe.
- **Anaglyph** – Metoda s barevným rozdělením obrazů.
- **Interlaced** – Horizontální prokládání řádků.

- **Above-Below** – Metoda se zobrazením obrazů nad sebou.
- **Quad-Buffer** – Metoda, která obrazy střídavě zobrazuje s vyšší frekvencí.

Většina těchto zobrazení bude podrobněji rozepsána v následujících kapitolách. Potíž s touto implementací stereoprojekce je ta, že její využití je možné pouze při spuštění „game módu“, případně spuštění přes přiložený Blender player. Použitelnost tohoto sterea je tedy pouze mimo 3D Viewport, ve kterém uživatelé pracují a modelují. Nová verze engine byla již navíc už s myšlenkou stereoprojekce navrhována a implementována, tudíž nebylo potřeba žádných zásadních úprav pro to, aby ji mohl uživatel využívat. Implementace samotná je také mnohem jednodušší, jelikož se zde nemusí počítat s vykreslováním hran, vertexů, případně různých popisků nebo jiných objektů scény, které se v ní vyskytují (světla, kamera, řídicí mřížka a podobné).

2.3 Blender player

Tato část programu Blender byla vytvořena pro spouštění zmíněných interaktivních prohlídek nebo her. Přehrávač využívá zobrazovacích možností Blenderu s tím, že zde chybí veškeré GUI a editační prvky. Tímto zredukováním bylo dosaženo toho, že si uživatel může sám své GUI a ovládání scény vytvořit pomocí skriptů v jazyce Python. Přehrávač umí rovněž zobrazovat scénu se stereoprojekcí podobně jako Game Engine. Toto zobrazení přímo využívá Game Engine, nejedná se tedy o oddělenou implementaci stereoprojekce. Jedna z výhod přehrávače je ta, že existuje spousta možností v nastaveních před spuštěním. Oproti testování interaktivních aplikací přímo v Blenderu lze například spustit aplikaci ve fullscreen módu, který některé druhy sterea přímo vyžadují. Dalšími možnostmi jsou například zadání rozlišení obrazovky, druh stereoprojekce nebo zobrazovací frekvence.

3 Prostorové vidění

Lidé, stejně jako spousta dravců, mají vlastnost vnímat hloubku okolí. Lidské vnímání světa a jeho hloubky prostřednictvím očí je založeno na poměrně komplexním principu. Existuje celá řada faktorů, které náš mozek vyhodnocuje předtím, než je schopen určit celou hloubku scény. Ale právě díky odsazenosti očí dokážeme pro náš mozek získat dva nepatrně posunuté a pootočené obrazy scény před námi. Díky těmto dvěma snímkům je pak náš mozek schopen vypočítat hloubku vnímaných objektů. Při hloubkovém vnímání jsme schopni rozeznávat přibližné vzdálenosti a rozdíly ve velikostech různých objektů[5].

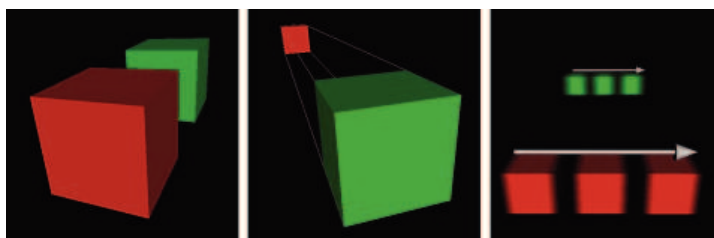
Hloubku samotnou člověk dokáže vnímat pomocí mnoha dalších vodítek. Některá z těchto vodítek lze vyčíst i z nestereoskopických 2D snímků (například fotografie, malby, filmy a podobně). V takových případech se jedná převážně o vnímání tzv. „naučené hloubky“. Tedy takové hloubky, o které jsme se dozvěděli až s postupným získáváním vlastních zkušeností. Toto vnímání hloubky je možné se naučit i s určitou poruchou prostorového vnímání[28].

Tyto a podobné vlivy se dají zařadit mezi psychologická, nebo také monokulární vodítka. Některé z těchto vodítek pro získávání prostorového dojmu (i z 2D snímků) jsou tyto:

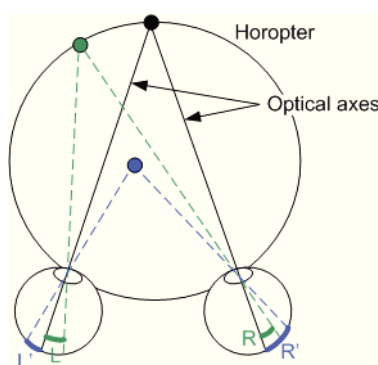
- **Lineární perspektiva** – Čím jsou objekty dále, tím se nám zdají menší. Paralelní linie konvergují v dálce.
- **Atmosférická perspektiva** – Vzduch, mlha a další atmosférické vlastnosti okolí zkreslují vidění vzdálenějších objektů. Díky takovému zkreslení jsme pak schopni odhadnout jejich vzdálenost.
- **Velikost objektů** – Jsou-li tyto objekty pro nás známé, jsme schopni určit v jaké přibližné vzdálenosti se nacházejí. Pokud například na fotografii uvidíme automobil, který se jeví stejně velký jako pes, pak je nám jasné, že pes je k nám vůči automobilu umístěn mnohem blíže.
- **Detail** – Blíže objekty se nám zdají detailnější než ty vzdálené.
- **Překrývání** – Předpokládáme, že pokud nějaký objekt překrývá jiný, pak se nachází v popředí.
- **Světla a stíny** – Blíže umístěné objekty se nám zdají světlejší než objekty v dálce. Pomocí světla a stínů jsme také schopni rozeznávat nepravidelnost nebo zakřivení povrchů.
- **Relativní pohyb** – Objekty v dálce se jeví jako pomalejší než ty v popředí.

Existují však i podmínky, které žádný nestereoskopický 2D snímek nabídnout nedokáže. Jedná se o fyziologická, nebo také binokulární vodítka, která jsou možná pouze zapojením obou očí. Mezi tyto vodítka patří:

- **Binokulární disparita** – Tato vlastnost je základem pro tvorbu každého stereoskopického snímku. Jedná se o jev, kdy dva obrazy světa, vytvořené párem očí, nejsou stejné. Disparita je dána prostorovou diferencí umístění obou očí (u člověka se jedná v průměru o 63 mm)[20]. Pokud se zaměříme na jeden bod, pak se nám tento bod promítne na sítnice obou očí do určitého místa. Každý další bod, který se v obou očích promítne do stejné vzdálenosti od průmětu prvního bodu, leží na horopteru a má nulovou disparitu. Pro nás se takový bod jeví stejně daleko jako je bod první. Laicky řečeno, z drobných rozdílů obrazů vnímaných v obou očích lze rekonstruovat hloubkové poměry ve scéně[28].
- **Akomodace** – Schopnost oka zaostřit pomocí oční čočky. Oči v prostoru zaostřují na různě vzdálené objekty[28].
- **Konvervence** – Jedná se o stav, kdy zaměříme naše oční osy na určitý objekt. Pokud se snažíme dívat na blízký objekt, pak k němu tyto osy konvergují (sbíhají). Díváme-li se na objekty v dálce, pak jsou oční osy rovnoběžné. Mimořádně může docházet i k divergenci (rozbíhání) očních os, což je jev, který by neměl zdravému jedinci nastávat[28].



Obrázek 2: Překrývání, lineární perspektiva, relativní pohyb[23]



Obrázek 3: Binokulární disparita[23]

Z tohoto širokého výčtu veškerých vodítek je patrné, že může být složité uměle navodit všechny tyto vlivy, které mozek zpracovává tak, aby mohl plnohodnotně vyhodnotit hloubku scény. Samotná stereoskopie se zakládá převážně na využití binokulární disparity, která je pro toto vnímání nejdůležitější (a zároveň nejsnáze navoditelná). Při tvorbě stereoskopických scén je pak vhodné dodržovat pravidla uvedená v části o psychologických vlivech (například překrývání a podobně).

3.1 Stereoslepota

Stereoslepota je stav, kdy lidé nedokáží vnímat a vidět stereoskopické zobrazení. Jedná se o neschopnost vnímat hloubku určité scény kombinací dvou jejích snímků. Tento stav mohou mít jedinci s tupozrakostí nebo výrazným šilháním. Lidé s pouze jedním okem tímto stavem trpí neustále. Může nastat situace, kdy se z neznámých důvodů stereoslepota projeví i u zdravého jedince. Lidé, kteří trpí tímto stavem, dokáží vnímat hloubku okolí pouze skrze monokulární vodítka.

3.2 Zásady pro vytváření stereoskopických snímků

Vytvořit stereoskopický snímek s pouhým posunutím dvou kamer, je základ. Jsou zde ale i další aspekty, které by měl tvůrce dodržet tak, aby mohl divákovi zajistit, co možná nejpříjemnější podmínky pro sledování. Existují aspekty, které když nebudou při této tvorbě dodrženy, mozek nebude schopen uvědomit si prostorovost a hloubku objektu. Při nedodržení těchto zásad vznikají nepříjemné situace, které mohou zkreslit celkový dojem vnímané hloubky. Bohužel ne všechny tyto podmínky je tvůrce takového zobrazení schopen dodržet, protože mnoho z nich závisí i na vnějších vlivech, nebo nastalých situacích[9].

- **Ghosting** – Nepříjemný jev, kdy je obraz určený pro jedno oko viděn i druhým okem. Může to být také stav, kdy si mozek není schopen uvědomit, že dva vnímané obrazy patří jednomu objektu. Druh a zřetelnost ghostingu záleží na využití zobrazovací metodě. Může se například stát, že při použití metody anaglyph nejsou zobrazované barvy dokonale shodné s barvami využívaných brýlí, tudíž dokážeme vidět náznak obrazu i okem, pro které nebyl určen. Ghostingu je možné dosáhnout také ve chvíli, kdy se objekty přiblíží k divákovi natolik, že jejich rozteč začíná být pro mozek neúnosně velká. V tuto chvíli mozek přestane pár obrazů chápat jako snímky jednoho objektu.
- **Kontrast** – Úroveň ghostingu u metody anaglyph se dá snížit zvolením správného kontrastu a vyváženosti barev.
- **Ohraničení obrazovky** – Důležitou podmínkou pro stereoskopické zobrazování je vhodná projekční plocha. Máme-li malou projekční plochu, stereoskopické zobrazení sice uvidíme, ale bude těžší jej naplno prožít. Je důležité rozhodnout, zda chceme divákovi zobrazit obraz vystupující z projekční plochy, nebo jestli chceme vykreslit objekty spíše do hloubky zobrazované scény. Pokud se snažíme donutit

scénu jakoby „vystoupit“, pak si musíme uvědomit, že objekty, které se nacházejí jakoby vně projekční plochy, mohou být na okrajích ořezány jejím ohrazením. Toto ohrazení se pro naše oči nachází až za touto scénou, tudíž se jedná o velmi nepříjemný jev.

- **Překrytí scény náhodnými prvky** – Prohlížení stereoskopické scény, která se jeví „vystouplá“ ven z projekční plochy, by mohlo být narušeno například průchodem osoby, nebo jakoukoliv věcí blokující divákův výhled. Podobně jako ohrazení obrazovky může být tento předmět vnímán jako vzdálenější než objekt zobrazované scény, a přesto bychom jej uviděli na popředí.
- **Strukturální interference** – Může nastat situace, kdy se v zobrazované scéně bude nacházet nějaký opakující se vzor, který má rozteč stejnou jako je ta stereoskopická. Vzniklá situace může vést k nechtěným a neurčitým artefaktům, které namáhají oči a zneprůjemňují celkový vzhled scény. Proto je vhodné se takovým opakujícím se vzorům vyhýbat, případně zajistit jejich odlišnou horizontální rozteč vůči té stereoskopické.
- **Zrnité textury** – Pokud se ve scéně nachází zrnité textury (například fotografie trávníku), které mají svou zrnitost mnohem vyšší, než je horizontální rozteč stereoskopických obrazů, vzniká obdobný jev jako u předchozího bodu.
- **Zrcadlení a odlesky** – Zrcadlení a odlesky se počítají vzhledem k úhlu mezi zdrojem světla a kamerou, která snímá scénu. Pokud se tyto obrazy s posunem kamery nevypočítají korektně, může docházet k nepříjemným efektům a namáhání očí. Existují metody, které odlesky mohou počítat pouze přibližně pomocí aproximací. Odlesky se jeví jako bezproblémové, pokud není využito stereoskopie, avšak s využitím sterea mohou v takto tvořených obrazech vznikat rozdíly. Tyto rozdíly pak mohou snižovat prostorový dojem.

3.3 Paralaxa

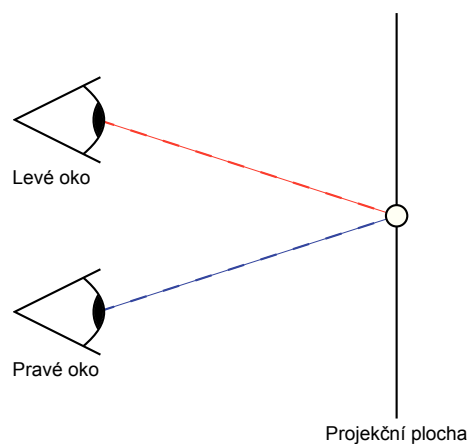
Jak bylo řečeno dříve, pro získání prostorového vjemu je důležité mít dva od sebe horizontálně odsazené obrazy. Paralaxa pak určuje velikost tohoto odsazení. Paralaxa může být definována jako vzdálenost mezi těmito dvěma obrazy, nebo jako úhel, který při pohledu na tyto obrazy svírají osy našich očí[10]. Pomocí paralaxy je také možné určit, zda-li se vykreslovaný objekt zobrazuje před projekční plochou, nebo za ní.

Jsou známé čtyři druhy paralaxy. Některé z těchto paralax jsou vhodné pro určité situace a jiné by neměly nastávat vůbec[29].

3.3.1 Nulová paralaxa

O nulové paralaxe mluvíme ve chvíli, kdy objekt našeho zájmu leží přesně na projekční ploše. Umístění takového objektu se divákovi nejeví v popředí, ani v pozadí. Nulovou paralaxu lze získat tak, že oba obrazy určitého objektu splynou v jeden (mezi nimi je

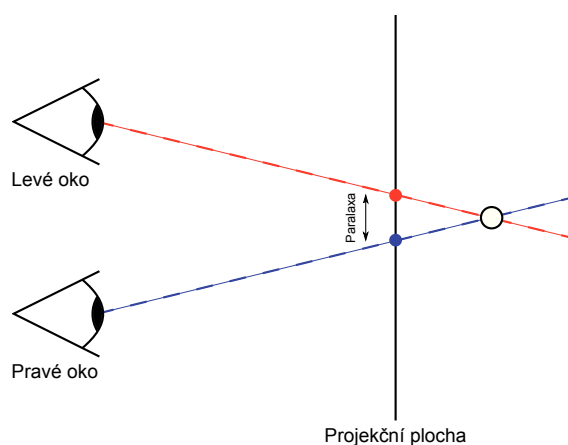
nulová vzdálenost). Dá se říci, že tato paralaxa je běžná při prohlížení nestereoskopických snímků. Na obrázku je patrné, že oči konvergují ve stejném bodě umístěném právě na projekční ploše, díky čemuž nevznikají žádné dva odsazené obrazy.



Obrázek 4: Nulová paralaxa

3.3.2 Pozitivní paralaxa

Pozitivní paralaxa je případ, kdy se zobrazovaný objekt nebo scéna jeví jakoby za projekční plochou. Této paralaxy lze dosáhnout ve chvíli, kdy se obraz pro levé oko zobrazuje vlevo a obraz pro pravé oko vpravo. Tím, že jsou tyto obrazy na své straně, je jejich horizontální odsazení kladné → pozitivní paralaxa.

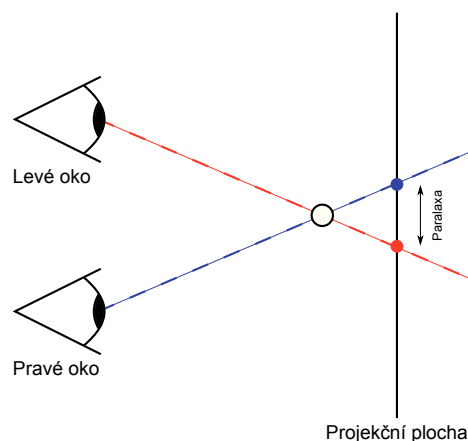


Obrázek 5: Pozitivní paralaxa

Maximální horizontální paralaxy lze dosáhnout ve chvíli, kdy je zobrazovaný objekt v nekonečnu a osy našich očí jsou rovnoběžné. Jednoduše řečeno, čím blíže se objekt nachází k projekční ploše, tím se jeho paralaxa zmenšuje, až nakonec oba obrazy splynou. Ve chvíli, kdy oba zobrazované obrazy splynou, je vykreslovaný objekt k divákovi nejbližší a nachází se na projekční ploše. Při posunu objektu dál od projekční plochy se rozestup zobrazovaných obrazů zvětšuje. Z obrázku lze vyčíst, kde se na projekční ploše promítají obrazy objektu divákova zájmu. Je vidět, že tyto promítané obrazy jsou na stejné straně jako oko, pro které jsou vykreslovány. Je zároveň patrné, ve kterém místě se výsledný objekt nachází.

3.3.3 Negativní paralaxa

Negativní paralaxa nastává ve chvíli, kdy se objekt jeví jakoby před projekční plochou. Při negativní paralaxě se osy našich očí setkávají ještě před projekční plochou. Této paralaxy je docíleno tím, že se obrazy pro jednotlivé oči na projekční ploše prohodí. Prohozením obrazů máme na mysli to, že obraz pro levé oko se nachází napravo od obrazu pro pravé oko.

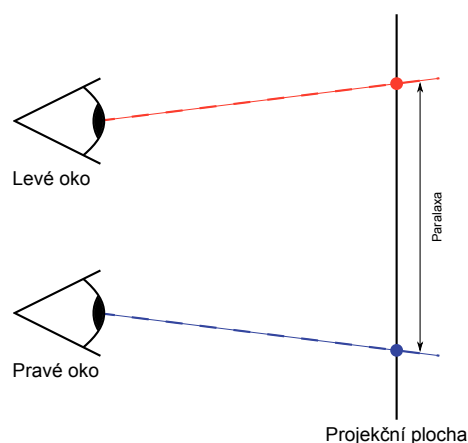


Obrázek 6: Negativní paralaxa

Oproti pozitivní paralaxě se při vzdalování objektu od diváka jeho horizontální odsazení zmenšuje. Ve chvíli, kdy oba obrazy splynou, se daný objekt nachází na projekční ploše a je divákovi vzdálen nejvíce. Pokud se objekt blíží k divákovi, pak roste i jeho horizontální odsazení. Je-li objekt k očím diváka přiblížen přes určitou kritickou mez, pak se začne rozdvajovat. V důsledku toho mozek přestává obrazy vnímat jako jeden celek a vzniká ghosting. Na obrázku je vidět, jakým způsobem jsou prohozeny vykreslované obrazy. Je patrné i místo, kde divák vidí zobrazovaný objekt vůči projekční ploše.

3.3.4 Divergentní paralaxa

Divergentní paralaxa nastává ve chvíli, kdy osy našich očí přestávají být rovnoběžné a začínají divergovat. Tento stav je pro lidské vnímání nepřírozený a může se projevovat zvýšenou námahou očí. Divergentní paralaxa může nastat ve chvíli, kdy se zobrazovaným obrazům nastaví nepřiměřeně velké horizontální odsazení. Při stereoskopické projekci se tomuto stavu snažíme vždy vyhnout. Z obrázku je patrné, že díky divergenci očních os nikdy nemůže nastat situace, kdy budou vykreslované obrazy určovat jeden objekt (osy se nikdy nezkříží).



Obrázek 7: Divergentní paralaxa

3.4 Metody získávání stereopárů

Co je to vlastně stereopár? Tento název se využívá pro soubor dvou obrazů získaných horizontálním posunem nějakých nahrávacích jednotek (kamer, fotoaparátů atd.). Jsou dány tři základní metody určující způsob umístění a natočení těchto aparátů. Následně představené metody lze využít jak při reálném nastavování kamer a fotoaparátů, tak při nastavení virtuálních kamer v programu Blender[35].

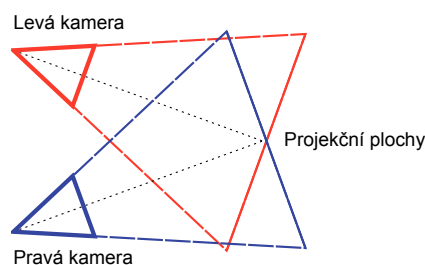
3.4.1 Toe-in

Tato metoda má nejbližší k lidskému reálnému vnímání objektů. Při použití této metody jsou osy obou kamer namířeny na jeden bod (objekt našeho zájmu). Když se v reálu snažíme zaostřit na určitý objekt, pak osy našich očí konvergují až k němu. Málokdy nastane situace, při které jsou osy našich očí zcela rovnoběžné tak, že nevnímáme žádný konkrétní objekt, ale celou scénu.

Problém s využitím této metody je ten, že tímto zaměřením na jeden objekt se zkresluje zbytek scény. Dochází k vytvoření tzv. vertikální paralaxy. Výsledný stereopár sice dokáže navodit pocit hloubky a je využitelný ve stereoskopii, ale pokud se divák zaměří

na body vzdálenější od objektu našeho zájmu, na který bylo zaostřeno, začne vznikat nepříjemné zkreslení, které způsobuje namáhání očí.

Tato metoda by byla nejlépe využitelná ve chvíli, kdy bude naprostá jistota, že se divák bude zaměřovat pouze na námi daný bod. Pokud by bylo možné v reálném čase zjistit, na který konkrétní objekt se divák v tuto chvíli zaměřil, a následně v závislosti na tom přesunout bod, ve kterém se protínají osy kamer, pak by tato metoda byla pro lidské oko perfektní a nejrealističtější.

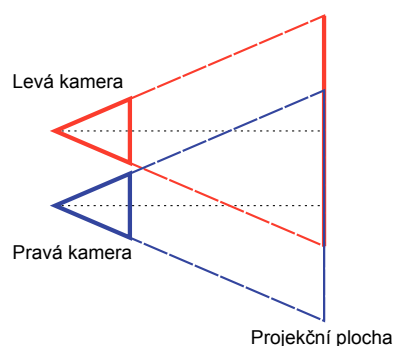


Obrázek 8: Toe-in metoda

3.4.2 Parallel

Využitím paralelní metody nevzniká vertikální paralaxa, tudíž nedochází k žádnému zkreslování scény. Osy obou kamer jsou spolu vzájemně rovnoběžné a nesbíhají se ke konkrétnímu bodu. Jedná se o jednu z dnes nejpoužívanějších metod. Právě díky rovnoběžnosti os obou kamer není divák nijak limitován zkreslením a může se zaměřit na kterýkoliv objekt ve scéně.

Jediným problémem této metody jsou levý a pravý okraj scény. Z obrázku je vidět, že tyto okraje jsou zabírány vždy pouze jednou kamerou, tudíž ztrácí své stereoskopické vlastnosti. Pokud se divák zaměří na tyto okraje, nebude to mít za následek zvýšenou námahu očí, ale jistou ztrátu hloubky. Tento problém řeší metoda, která je popsána níže.



Obrázek 9: Paralelní metoda

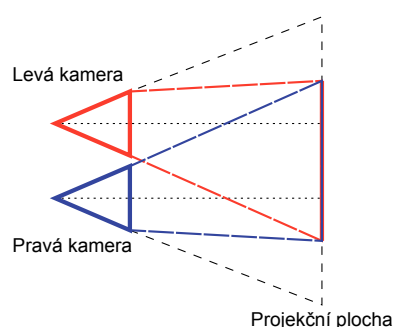
3.4.3 Off-axis

Tato metoda snímá obrazy podobně jako paralelní metoda popsaná v kapitole 3.4.2. Jsou zde dva horizontálně odsazené aparáty s rovnoběžnými osami sbíhajícími se v nekonečnu. Stejně jako u předchozí metody nevzniká žádná vertikální paralaxa, která by vedla k namáhání očí.

Rozdíl mezi off-axis a parallel metodou je ten, že off-axis metoda navíc ze získaných snímků vyřezává pouze tu oblast, která je společná pro obě oči. Tím na projekční ploše nevznikají žádné oblasti nebo artefakty narušující stereoskopickou projekci.

Tato metoda se zdá být pro stereoskopickou projekci nejlepší, jelikož nenavozuje žádné nepříjemné situace popsané v kapitolách 3.4.1 a 3.4.2, které by mohly zbytečně namáhat oči. Zajímavé je však to, že oproti předchozím metodám je tato nejméně podobná lidskému vnímání. Pokud se zamyslíme, pak si uvědomíme, že člověk většinou svůj zrak zaostřuje na konkrétní bod nebo objekt. Málokdy se stává, že osy našich očí jsou rovnoběžné. Dokonce náš mozek ani „neorezává“ získané obrazy způsobem, jakým zde bylo popsáno. Vždy se na okrajích našeho zorného pole vyskytují části, které vnímá pouze jedno oko a druhé nikoliv.

Výhoda této metody je ta, že nezáleží na momentálních požadavcích diváka. Celá scéna je vykreslena tak, aby vyhovovala všem bez nutnosti určovat, který bod je právě objektem zájmu a které části obrazu jsou pro diváka periferní.



Obrázek 10: Off-axis metoda

4 Stereoskopická projekce

Stereoskopická projekce se snaží vytvářet promítaný obraz takovým způsobem, aby byl svým publikem vnímán prostorově. Tato metoda je založena na principu promítání dvou obrazů tak, aby každý z nich byl viděn pouze okem, pro které byl zamýšlen. Rozdíly v promítaných obrazech jsou právě v jejich horizontálním odsazení. Je mnoho způsobů, pomocí kterých určíme, který snímek se zobrazí pro dané oko, ale obecně lze tyto techniky rozdělit na pasivní a aktivní. V další části jsou popsány některé vybrané způsoby stereoprojekce. Existuje více způsobů, kterými lze u diváků navodit dojem hloubky. Tyto možnosti ale nebudou blíže rozepsány, jelikož přímo nesouvisí s popisovanou úpravou programu Blender.

4.1 Pasivní projekce

Pasivní projekce se zakládá na různých principech, které budou podrobněji popsány níže. Všechny použité principy mají společné to, že pro rozdělení očí jsou využity pasivní brýle, které mají své dané vlastnosti. Pasivní brýle není potřeba nabíjet nebo je složité synchronizovat s projekční plochou. Výhodou této metody je určitě lehčí váha brýlí, jejich pořizovací cena a oproti některým aktivním brýlím také to, že zde není přítomno žádné nepříjemné blikání.

4.1.1 Anaglyph

Nejjednodušší možnost. Tato metoda využívá pro oddělení snímků barevné filtry. Zde není potřeba žádného speciálního hardwaru ani drahých brýlí. Anaglyph brýle mají barevně rozlišená jednotlivá skla tak, aby vždy propustila jen snímky ve správné barvě. Je známo více druhů barevných rozdělení, ale nejčastěji se dnes využívají červeno-azurové (kde je červený filtr na levém oku a azurový na pravém). Známá jsou i jiná barevná rozdělení jako například červeno-modré, žluto-modré nebo červeno-zelené.

Princip této metody tkví v tom, že vykreslované horizontálně posunuté snímky jsou tvořeny barvami, které jsou vzájemně doplňkové. Na příkladu je tato metoda uvedena s použitím barev červená-azurová. Obrazy pro jedno oko jsou převedeny do azurové barvy a obrázky pro druhé oko do červené barvy. Tyto obrazy se pak překreslí vzájemně přes sebe. V místech, kde se odstíny těchto dvou barev setkávají, se dá získat barva původního obrazu. Tam, kde došlo k posunu, jsou jasně viditelné červené, případně azurové kontury.

Výhody této metody jsou její nízká cena a dostupnost. Nevýhodou představuje menší kvalita snímků. Dochází také ke zkreslení nebo dokonce úplnému potlačení některých barev. Zároveň je zde možnost vzniku tzv. „duchů“.

4.1.2 Polarizace

Novější metoda, která pracuje na principu polarizace světla. Tato metoda se snaží oddělit obrazy pro každé oko tak, že obraz pro jedno oko je polarizován horizontálně a obraz pro druhé oko vertikálně.

U některých moderních televizorů a monitorů není při použití této metody, oproti aktivní projekci, potřeba vyšší zobrazovací frekvence. Zobrazování polarizovaných snímků se provádí pomocí horizontálního (případně vertikálního) prokládání. Při horizontálním prokládání se každý lichý řádek polarizuje pro jedno oko a každý sudý řádek pro druhé oko. Pokud je využito vertikální prokládání, pak se polarizují jednotlivé zobrazované sloupce. Z popisu této metody jasně vyplývá, že zobrazované stereoskopické obrazy jsou v určitém směru o polovičním rozlišení než ty původní.

Lepší využití této metody se nabízí, pokud je namísto monitoru použita soustava dvou projektorů, které promítají obrazy přes sebe. Jeden projektor promítá polarizované snímky s plným rozlišením pro jedno oko a druhý projektor pro druhé oko. Tyto projektory pak musí obraz promítat na nepolarizační (takovou, která nezmění polaritu zobrazovaných snímků) projekční plochu. Tato metoda je však pro soukromé využívání poměrně nákladná.

4.2 Aktivní projekce

U této metody zobrazování se využívá aktivních brýlí. Brýle musí obsahovat vlastní elektroniku a bývají obvykle těžší než brýle pasivní. Aktivní brýle fungují na principu clon tvořených tekutými krystaly, které mají dva stavy. V jednom stavu jsou tyto krystaly ztmaveny a nejsou schopné propouštět světlo ani obraz, tudíž přes ně nelze nic vidět. Ve druhém stavu jsou tyto krystaly propustné a promítaný obraz je viditelný. Máme-li pak takové brýle, které mají pro každé oko tuto zvláštní clonu, jsme schopni určovat, které oko bude v určitou chvíli na projekční plochu vidět a které ne. Takové brýle je nutné synchronizovat s projekční plochou tak, aby bylo aktivní vždy to oko, pro které se promítá obraz. Nevýhodou takových brýlí může být například i to, že mírně snižují jas a kontrast zobrazovaného obrazu.

Projekční plocha by při využití této metody měla mít zpravidla vyšší zobrazovací frekvenci (dnes obvykle 120 Hz). Díky vyšší frekvenci jsme pak schopni vysílat střídavě snímky pro každé oko o poloviční frekvenci (60 Hz) bez znatelnějšího blikání. Výhodou tohoto vysílání je, že je možné bez problému docílit Full HD rozlišení. Nepříjemné blikání se dnes někteří výrobci snaží zredukovat ještě větším zvyšováním zobrazovací frekvence (až 400 Hz). Důležité je také osvětlení místnosti. Při větším zatemnění místnosti dochází k podstatné redukci vnímatelného blikání.

4.3 Autostereoskopický displej

Metoda s využitím autostereoskopického displeje je oproti předcházejícím metodám jiná hlavně v tom, že zde divák nepotřebuje žádné speciální brýle. Nejsilnější stránkou této

metody je právě zmiňovaná absence brýlí. Negativem je zhoršení pozorovacích úhlů, ve kterých jsou stereosnímky korektně promítány pro oči, pro které jsou zamýšleny.

Společným znakem všech autostereoskopických displejů je speciální maska umístěná před samotným displejem. Tato maska je vybavena optickými hranoly, které vychylují různé sloupce (případně řádky) pixelů do různých směrů. Na obrazovce monitoru jsou podle parametrů optické masky vtěsnány obrazy pro pravé i levé oko. Maska způsobí to, že jsou obrazy pro jednotlivé oči viditelné pouze z určitých směrů. Pokud se tedy divák postaví do vhodné pozice (pro každého člověka může být jiná – to podle rozteče očí), uvidí levým okem pouze levý obraz a pravým pravý. Další nevýhodou této metody je to, že rozlišení zobrazovaných obrazů je poloviční než rozlišení projekčního zařízení[3].

Způsoby tohoto zobrazování mohou využívat různých druhů masek. Nejznámější využívané masky jsou parallaxová bariéra nebo lentikulární maska. Pro tento způsob stereoprojekce bylo vytvořeno více různých metod, které vylepšují a rozšiřují možnosti pozorovacích úhlů. Můžeme narazit například na eye-tracking metodu, která se snaží pomocí kamery zaměřit pozici divákových očí a fyzicky posunout stereoskopickou masku tak, aby vyhovovala uživateli. Použitelnost této metody je bohužel snížena ve chvíli, kdy se snímky promítají pro více diváků najednou. V dnešní době existují i autostereoskopické systémy s pozorovacím úhlem 360 stupňů, kdy je možné zobrazovanou scénu obcházet a pozorovat ze všech stran.

5 Implementace

Vzniklá úprava programu Blender je vystavěna na původní verzi 2.62.3. Konkrétně se jedná o SVN revizi číslo 45345. Revize používaných knihoven, které program Blender vyžaduje ke svému přeložení, je 45112. Způsobů, jakými lze Blender získat je několik (např. SVN, stažení aktuální verze z oficiálních stránek[6] apod.). Vzhledem k potřebě neustálých průběžných aktualizací verze, se kterou programátor pracuje, se jako nejvýhodnější možnost jeví využití SVN. Pomocí SVN je rovněž možné bez větších obtíží spojit současnou vlastní stereoskopickou úpravu programu Blender s jeho novější verzí. Díky těmto kladům bylo SVN zvoleno jako způsob, kterým je pracovní kopie programu Blender stažena a následně aktualizována. Na oficiálních stránkách se nachází podrobný návod jak pomocí SVN získat poslední verzi programu Blender[7]. Postup získání a aktualizace zdrojových kódů programu Blender byl tento:

- **Využitý software** – SVN program, kterým byly aktualizace prováděny se nazývá TortoiseSVN. Pro vygenerování projektových souborů byl použit program CMake. Jako pracovní prostředí a kompilátor bylo využito Visual Studio 9 2008.
- **Checkout** – Pravým tlačítkem myši klikneme na složku (např. C:\BlenderSVN), která bude reprezentovat úložiště pro pracovní kopii námi stahovaného programu. V menu pak vybereme možnost *SVN checkout*. V následně vyvolaném okně se do pole URL vloží adresa: `https://svn.blender.org/svnroot/bf-blender/trunk/blender`. Nakonec potvrdíme vybranou složku (např. C:\BlenderSVN\blender), do které se zdrojové soubory stáhnou. Celý tento postup zopakujeme pro stažení potřebných knihoven. V tomto případě je do pole URL zapsána adresa: `https://svn.blender.org/svnroot/bf-blender/trunk/lib/windows`. Opět potvrdíme umístění složky (např. C:\BlenderSVN\lib\windows), kde se budou tyto knihovny nacházet.
- **Update** – Pokud si přejeme námi staženou pracovní kopii aktualizovat, pak stačí pravým tlačítkem myši kliknout na vytvořenou složku (např. C:\BlenderSVN\blender, nebo C:\BlenderSVN\lib\windows) a ve vyvolaném menu zvolit možnost *SVN update*.
- **Generování** – Pro vygenerování potřebných projektových souborů pomocí programu CMake je nutné jako zdrojovou složku zvolit tu, která obsahuje stažené soubory (v našem případě se jedná o složku C:\BlenderSVN\blender). Cílovou složku je možné zvolit podle potřeby (např. C:\BlenderSVN\build). Po zadání potřebných cest stiskneme tlačítko *configure*. Ve vyvolaném menu vybereme cílový kompilační systém (např. Visual Studio 9 2008). Opětovně stiskneme tlačítko *configure*. Po druhé konfiguraci pak stačí stisknout tlačítko *generate*. Tímto jsou vygenerovány všechny potřebné soubory.
- **Kompilace** – Pro překlad a vytvoření spustitelného souboru s využitím programu Visual Studio 9 2008 je vždy důležité nejprve zkompilovat část s názvem *INSTALL*.

Po přeložení této části jsou do složky se spustitelným souborem přidány potřebné knihovny, bez kterých program samotný spustit nelze. Pro přeložení ostatních souborů a vytvoření finálního spustitelného programu je pak nutné zkompileovat část *ALL_BUILD*. Jakmile jsou obě tyto části zkompileovány, je vytvořena plnohodnotná a spustitelná verze programu Blender.

Před přistoupením k samotnému přepisu některých částí zdrojových kódů bylo nutné si vytvořit určité povědomí o jejich architektuře a způsobech zpracovávání některých požadavků. Nebylo potřeba procházet všechny části programu Blender, jelikož se naše pozornost soustředí převážně na vykreslovací metody. Vizuální zpracování této architektury je znázorněno na diagramu umístěném v příloze A. Na následujících stránkách budou popsány ty části programu, kterých se prováděné stereoskopické úpravy týkaly.

5.1 Creator

V této části programu Blender se nachází hlavní metoda *main*. V této metodě se také inicializuje mnoho dalších metod a procesů. Zde byla také umístěna inicializace stereoprojekce. Soubor *creator* zároveň zaštiťuje zpracovávání přidáných parametrů při spouštění aplikace.

Na počátku tvorby této diplomové práce byly pro snazší spouštění vytvořeny určité pomocné parametry, kterými uživatel mohl před spuštěním pohodlně nastavit způsob a rozteč stereoprojekce. Tato metoda nastavování zůstala navzdory tomu, že byl později do samotného GUI přidán ovládací panel. Kromě nastavení samotné stereoprojekce se v metodě *main* nachází i funkce pro zpracovávání určitých spouštěcích parametrů. Nově definované parametry bylo potřeba vytvořit následujícím způsobem:

- **Registrace** – Je nutné registrovat název a způsob volání nově vzniklého parametru. Zároveň se k tomuto parametru při registraci přiřazuje metoda, která daný parametr zpracovává.
- **Definice metody** – Před samotnou registrací musí být vytvořena metoda, která je danému parametru přiřazena. V této metodě pak probíhá zpracování a vyhodnocení přidáného parametru.

Ukázka registrace spouštěcích parametrů a jejich zpracování pomocí nově přidáných funkcí je předvedena v následujícím výpisu zdrojového kódu.

```
static int set_hwpagflip(int argc, const char **argv, void *data)
{
    if (argc >= 1)
    {
        float f = atof(argv[1]);
        myEye = f;
        myStereoState = 1;

        return 1;
    }
}
```

```

    else
    {
        return 0;
    }
}
...
BLI_argsAdd(ba, 4, "--hwpageflip", "--hwpageflip", "hwpageflip", set_hwpageflip, NULL);

```

Výpis 1: Registrace a zpracování parametrů

Podobným způsobem, jaký byl znázorněn na výše uvedeném zdrojovém kódu, byly zaregistrovány a zpracovány také ostatní parametry. Získané hodnoty se ukládají do globálních proměnných, které se využívají při inicializaci stereoprojekce v metodě *main*.

V metodě *main* se při spuštění programu Blender stereoprojekce inicializuje pomocí funkce *eye.set*, která zpracovává získané hodnoty, nastavené uživatelem. Takto získané hodnoty se pak ukládají do struktury *bContext*. Po inicializaci stereoprojekce se dále volá metoda *WM_main*, která spravuje veškerá vytvořená okna a menu programu Blender.

5.2 Context

Soubor *context* obsahuje strukturu *bContext*, která je využívána téměř v celém zdrojovém kódu. Tato struktura obsahuje různá nastavení a důležité proměnné. Důvodem vytvoření této struktury bylo pravděpodobně sjednocené spravování jinak oddělených důležitých proměnných, ke kterým bývá vyžadován častý přístup v mnoha místech zdrojového kódu. Díky tomuto sjednocení stačí mezi jednotlivými metodami a funkcemi přeposílat ukazatel na tuto strukturu. Samotná struktura je definována a inicializována uvnitř metody *main* umístěné v souboru *creator*.

Právě díky „všudypřítomnosti“ této struktury byl *bContext* zvolen jako nositel informací o přidáných stereoskopických nastaveních. Díky této přenosné struktuře nebylo potřeba vytvářet singleton, nebo vlastní obdobnou strukturu, kterou by bylo možné přeposílat mezi jednotlivými metodami.

Do struktury *bContext* byla přidána další struktura *ms* (zkratka my stereo), ve které jsou definovány nově vzniklé stereoskopické proměnné. Soubor *context* také obsahuje různé metody, pomocí kterých bývají uložená data zpracovávána. Bylo tedy vhodné umístit zde i takové předdefinované metody, které budou ukládat, zpracovávat a vracet potřebné stereoskopické proměnné.

Pomocí této struktury je možné přenášet veškerá důležitá stereoskopická nastavení napříč celou aplikací od její inicializace až k správci oken, zpracovávání klávesových zkratk nebo vykreslování. Přidaná struktura *ms* obsahuje následující prvky:

- **Stereoskopické proměnné** – Je zde umístěno několik proměnných, které reprezentují vybranou stereoskopickou metodu, rozteč mezi snímky nebo stav stereokurzu.
- **Prokládací masky** – Své místo zde mají také předdefinované masky, které bývají využívány při prokládané stereoprojekci.

- **Ukazatele na seznam oken** – Z určitých důvodů, které jsou popsány v kapitole věnované stereokurзору, je zde umístěn i seznam ohrazení jednotlivých vykreslovaných scén.

V níže uvedeném výpisu ze zdrojového kódu je znázorněno, jakým způsobem jsou v souboru *context* definovány metody, pomocí kterých je možné přistupovat k datům uloženým v přidané struktuře. Konkrétně se zde jedná o metody pro práci s proměnnou reprezentující stereoskopickou rozteč.

```
float getEyeDistance(const bContext *C)
{
    return C->ms.eyeDistnace;
}

void setEyeDistance(bContext *C, float dst)
{
    C->ms.eyeDistnace = dst;
}
```

Výpis 2: Práce s proměnnými

5.3 Window manager

Uvnitř souboru *wm* se nachází část window manager. V této části se spravuje většina aktivit spojených se zobrazovanými okny programu Blender. Konkrétně se v tomto souboru například inicializují nebo uvolňují různá vyvolávaná menu. V hlavní metodě tohoto souboru *WM_main* je umístěna nekonečná smyčka, která postupně zpracovává různé události, handlers, notifikace, aktualizace oken a pro tuto práci důležité vykreslování.

Skrze metodu *WM_main* se opět přeposílá ukazatel na strukturu *bContext* do všech jejích volaných částí uvnitř nekonečné smyčky. Touto cestou se všechny stereoskopické proměnné uložené v této struktuře přeposílají až ke stěžejní vykreslovací části. Vykreslování a aktualizace jednotlivých částí programu Blender se v této nekonečné smyčce provádí pomocí metody *wm_draw_update* definované v souboru *wm_draw*.

Uvnitř souboru *wm_draw* jsou umístěny důležité vykreslovací metody. Nachází se zde různé pomocné výpočetní funkce, které bývají volány při vykreslování některých částí aplikace. Důležitá je metoda *wm_draw_update*. V této metodě se provádí překreslování veškerého zobrazovaného obsahu aplikace. Zde se v závislosti na potřebě překreslení určité části rozhoduje, jak bude tato část vykreslena. V metodě *wm_draw_update* je také umístěno určité rozhodování o tom, které části musí být aktualizovány a překresleny. Bylo provedeno pár úprav tak, aby bylo docíleno, co nejlepší kompatibility se stereo-projekcí. Provedené úpravy byly integrovány tak, aby nebyla znehodnocena původní optimalizace vykreslování.

Úpravy, které jsou umístěny v této metodě, se týkají především ovlivnění požadavků na překreslení. Pokud je například stereoskopická scéna vykreslována pomocí metody *anaglyph*, není ovlivněna aktualizace této scény (scéna se aktualizuje a překreslí pouze při její změně – například při pohybu nebo rotaci). Pokud se ale scéna vykresluje metodou

využívající quad buffer, pak bylo důležité provést určité změny. Při vykreslování pomocí této metody jsou vysílány kompletní pozměněné obrazy pro obě oči, tudíž bylo zapotřebí vždy překreslit celou danou scénu. Kompletní překreslení scény zároveň probíhá, pokud dojde ke změně stávající stereoskopické rozteče.

V této části kódu se také nalézá prohození zobrazovacích bufferů. Při klasickém programování v OpenGL je nutné danou scénu po překreslení zobrazit. Toto zobrazování obvykle probíhá prohozením zobrazovacích bufferů pomocí příkazu *glutSwapBuffers*. Jelikož Blender knihovnu GLUT nevyužívá, je nutné toto překreslení provádět jinak. Prohození bufferů je zde vyvoláno pomocí příkazu *wm_window_swap_buffers*. Důvod, proč je zde toto prohazování zmíněno je ten, že je nezbytně nutné pro správné vykreslení scény.

Uvnitř této metody jsou také umístěny přidané funkce pro zpracování a vykreslení stereokurzu. Důvod, proč bylo vykreslení stereokurzu umístěno zde, namísto části programu věnované přímo vykreslování samotnému, je ten, že bylo nutné kurzor vykreslit až po celkovém překreslení scény. Zároveň díky správci oken, který se zde nachází, bylo možné zjistit momentální tvar, stav a pozici kurzoru. Právě díky tomuto správci pak nebylo obtížné daný kurzor jakkoliv měnit.

5.3.1 Stereokurzor

Při testování programu Blender se zabudovanou stereoskopickou úpravou si mnoho testovacích subjektů stěžovalo na nepohodlí při manipulaci s kurzorem. Zejména se jednalo o sníženou schopnost zaměřit požadovaný vertex případně linku. S klasickým kurzorem se uživatelům nejhůře zaměřovaly tenčí objekty, respektive takové, které byly vykreslovány s několika pixelovou šířkou.

Hlavní důvody tohoto nepohodlného zaměřování byly dva. Prvním velkým negativem byl fakt, že se uživatel snažil zaměřit trojrozměrný objekt pomocí dvojrozměrného kurzoru. Pojem trojrozměrný může být mírně zavádějící, jelikož to, že jsou vykreslovány modely trojrozměrné, se dalo říci i při vypnuté stereoprojekci. Rozdíl mezi trojrozměrností normálního modelu s modelem vykresleným pomocí stereoprojekce je, jak už bylo řečeno, ten, že stereoskopický model je tvořen pomocí tzv. stereopáru. Z důvodu vykreslení modelu pomocí dvou obrazů může být uživatel podvědomě zmaten. Nikdy nelze přesně určit, které oko bude při tomto míření dominovat, a tudíž na který ze dvou promítaných obrazů určitého bodu se uživatel zaměří. Při použití jakékoliv stereoprojekční metody se uživateli promítají dva obrazy, které se pouze jeví jako jeden stereoskopický snímek. To má za následek sníženou schopnost zaměřit uživatelem požadovaný bod, respektive jeho pravý obraz.

Druhým, uživateli často zmiňovaným negativním jevem, bylo rozdvojení kurzoru při zaměřování určitého stereoskopického bodu. Tento jev je dobře pochopitelný na příkladu s paralaxou. Uživatel se snaží zaměřit bod umístěný na objektu, který oproti kurzoru neleží na projekční ploše (v závislosti na paralaxe objekt leží buď před, nebo za projekční plochou). Při snaze uživatele sledovat takový objekt, dochází k situaci, kdy se osy jeho očí střetávají v jiné rovině, než se nachází projekční plocha. Tento střet os v jiné rovině, než je vykreslován kurzor, má za následek to, že jej uživatel vnímá dvojité.

Kompletním řešením obou problémů by bylo překreslení kurzoru na stereopár. Samotné překreslení není náročné, avšak bylo by důležité, aby se kurzor vždy nacházel ve stejné hloubce jako bod, na který je právě zaměřen. S touto metodou souvisí náročnější zjišťování, na kterém objektu se právě kurzor nachází, a s tím spojený výpočet přesného umístění zaměřeného bodu v prostoru. Tato metoda by mohla být při složitějších scénách a modelech výpočetně velmi náročná.

Jednodušším a mnohem rychlejším řešením se stal kurzor tvořený pouze polovičním stereopárem. Polovičním stereopárem je myšleno takové zobrazení, kdy je vytvořen obraz pouze pro jedno oko. Tak je zajištěno, že právě jedno oko vidí neposunutý vykreslený kurzor přesně v místě, kam v tuto chvíli míří. Obraz kurzoru se navíc promítá pouze pro to oko, které vidí reálný obraz scény a v ní obsažených objektů (takové, které lze označit nebo s nimi manipulovat). Pro druhé oko, se kurzor nepromítá. Díky tomu, že je kurzor viděn pouze jedním okem, je úplně potlačeno jeho rozdvajování při míření na stereoskopický objekt, který se nachází v jiné rovině než samotný kurzor. Toto řešení zároveň zmírňuje problémy při míření na správný obraz stereopáru zmiňované v předchozím odstavci.

Další logický krok, jenž by bylo vhodné vyřešit při rozšiřování této práce, by byl ten, který se snaží dořešit problémy vznikající se zaměřováním kurzoru v prostoru. Možnými cestami v dalším vývoji stereokurzoru by mohlo být využití specializovaného hardwaru (například myš s dalším kolečkem, pomocí kterého uživatel určuje hloubku, ve které se kurzor nachází), nebo vylepšení algoritmů, které pomocí výpočtů ve scéně určí hloubku kurzoru v závislosti na jeho pozici.

5.3.2 Implementace kurzoru

Samotná implementace stereokurzoru se skládala z několika částí. Jelikož nebylo potřeba mít tento kurzor vykreslený například při ovládání GUI nebo výběru v některém z pop-up menu, bylo důležité určit, ve které oblasti se kurzor bude překreslovat. V programu Blender je možné promítat různý počet zobrazení scény najednou. Tyto pohledy na scénu se navíc mohou nacházet v jednom, nebo i více oknech. Hranice každého zobrazení scény byly získány přímo při jejím překreslování. Díky tomu bylo možné tyto hranice uložit do struktury *ms* popsané v kapitole 5.2. Uvnitř této struktury se pak nachází seznam všech hranic právě vykreslovaných zobrazení. Jedná se o spojový seznam (linked list), kde každý prvek obsahuje kromě své hodnoty také ukazatel na následující prvek tohoto seznamu. Tento dynamický seznam je při každém překreslení scény naplněn, vyhodnocen a po zpracování výsledků opět vymazán tak, aby nedocházelo k přeplnění paměti, duplicitám nebo ke skladování neaktuálních dat.

Po vykreslení všech zobrazení dané scény je pak v závislosti na typu stereoprojekce vykreslen i tento kurzor. Vykreslování kurzoru probíhá v metodě *wm_draw_update*. Pokud je tedy uživatelem zadán požadavek na aktivaci stereokurzoru, pak se v této části provede kontrola, zda-li je kurzor uvnitř hranic scény a není otevřeno žádné pop-up menu. V případě, že je možné tento kurzor vykreslit, bude standardní kurzor zneviditelněn a na jeho místě se vykreslí stereokurzor. Vzhled kurzoru je pak upravený v závislosti na využívané stereoprojekční metodě.

Výsledný kurzor je upraven v závislosti na aktivní stereoprojekční metodě. Pokud je použita metoda využívající quad buffer, pak se kurzor vykresluje pouze do jednoho ze zadních bufferů. Je-li momentálně využívána některá z prokládaných metod, je na kurzor aplikována jedna z prokládacích masek.

5.4 Nastavení kamery

Při práci s programem Blender bylo nutné vybrat z několika možných úprav kamery tak, aby byl získán požadovaný stereopár. Nakonec bylo přistoupeno k co nejméně invazivní metodě, při které nebylo potřeba příliš zasahovat do původních zdrojových kódů.

Nejprve bylo nutné lokalizovat nastavení kamery, která snímá danou scénu. Uvnitř hlavní vykreslovací metody pro všechny objekty ve scéně *view3d_main_area_draw_objects* umístěné v souboru *view3d_draw*, je vyvolána krátká metoda, která provádí nastavování transformačních matic. Tato metoda byla upravena tak, že je z ní získán ukazatel na transformační matici, který je uložen uvnitř přidané globální proměnné. Proměnná je pak v závislosti na požadované rozteči upravována tak, aby docházelo k posunu vykreslovaných snímků. Ukázka úpravy matice je vidět zde:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p_x & 0 & 0 & 1 \end{bmatrix}$$

Kde T je transformační matice a p_x hodnota, o kterou se vykreslovaný obraz posune ve směru osy x [27].

Důležitý aspekt tohoto vykreslování je ten, že se transformace provádí pouze pro jedno oko. Jedno oko zůstává beze změny a druhé se posouvá o celou uživatelem zadanou rozteč. Tento způsob získávání stereopárů se nazývá asymetrický. Výhoda asymetrického promítání je ta, že se lépe přepíná mezi stereoskopickým a monoskopickým vykreslováním. Je-li potřeba promítat monoskopicky, stačí jen vypnout posunované oko.

Hlavní důvod, proč bylo rozhodnuto pro asymetrické promítání, byl i ten, že uživatel při modelování v programu Blender mnohdy pracuje jak s perspektivním, tak s ortogonálním vykreslováním scény. Při ortogonálním vykreslování není stereoprojekce tak žádoucím jevem jako u vykreslování v perspektivě. Pokud uživatel vyžaduje ortogonální promítání scény, jsou pro něj v tu chvíli důležitější informace o rovnoběžnosti a úhlech na úkor hloubky, o kterou se zde stereoskopie snaží. Pokud bychom při tomto druhu vykreslování scény nechali stereoskopii aktivní, uživatel by neviděl nic jiného než ploché objekty bez jakékoliv hloubky, které se jeví před (případně za) projekční plochou. Zde je tedy stereoskopie spíše na škodu, a tudíž bude deaktivována.

Druhým důvodem tohoto vykreslování je pozicování. Výhodnější je mít jeden z obrazů vykreslovaný tak, aby jednotlivé vertexy, plošky a linky odpovídaly svým reálným pozicím. Uživateli se pak lépe pracuje se scénou, jelikož bude schopen přesněji zaměřit kurzorem vykreslené části objektů. Toto zaměřování objektů souvisí zároveň s předchozí částí této kapitoly věnované stereokurzoru.

Přepoččet matice pro posunované oko uvnitř programu tedy probíhá tak, že se určí, která matice se právě upravuje. Následně přičteme zadanou rozteč k části, která reprezentuje posun po ose x . Nakonec je výsledná matice uložena zpět a aktualizována.

```
glMatrixMode(GL_MODELVIEW);
myRv3d->viewmat[3][0] += myStereoDistance;
glLoadMatrixf(myRv3d->viewmat);
```

Výpis 3: Úprava transformační matice

Samotné nastavování transformačních matic pak probíhá uvnitř cyklu. V tomto cyklu se rozhoduje, pro které oko se bude právě vykreslovaná scéna zobrazovat. Cyklus bude podrobněji popsán v následující části této kapitoly.

5.4.1 Využitá metoda

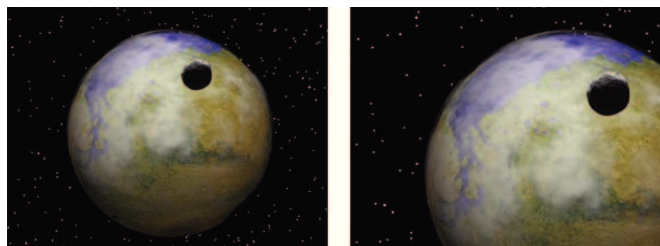
V kapitole věnované metodám získávání stereopárů byly zmíněny tři nejpoužívanější praktiky. Pro tvorbu stereoskopických obrazů v této úpravě programu Blender byla využita paralelní metoda. Paralelní metoda byla vybrána z čistě praktických důvodů. Oproti toe-in metodě se tato nezaměřovala pouze na jeden bod, ale komplexně na celou scénu. Tím, že jsou osy obou kamer vzájemně rovnoběžné, nedochází k žádnému zkreslování scény, nebo upřednostňování určitých objektů na úkor jiných. Může být namítnuto, že tato metoda neořezává pravý a levý okraj výsledného obrazu jako metoda off-axis. Off-axis je metoda, která své uplatnění nalezne například při pořizování stereoskopických fotografií, kde je odsazení snímacího aparátu mnohem znatelnější, díky čemuž vznikají i větší nestereoskopické „odřezky“.

Blender samotné vykreslování urychluje tím způsobem, že propočítává a vykresluje pouze ty objekty, které se vyskytují v momentálně viditelné části scény. Tím je zajištěno, že výpočty nebudou nijak znatelně zpomaleny ani při použití cyklu nutného pro získání stereopárů.

5.4.2 Parallaxa

Nemělo by smysl vytvářet tuto úpravu Blenderu, pokud by si uživatel přál nulovou parallaxu. Při nulové paralaxě by veškeré zobrazované obrazy stereopárů vzájemně splývaly a vykreslená scéna by se jevila jako nestereoskopická.

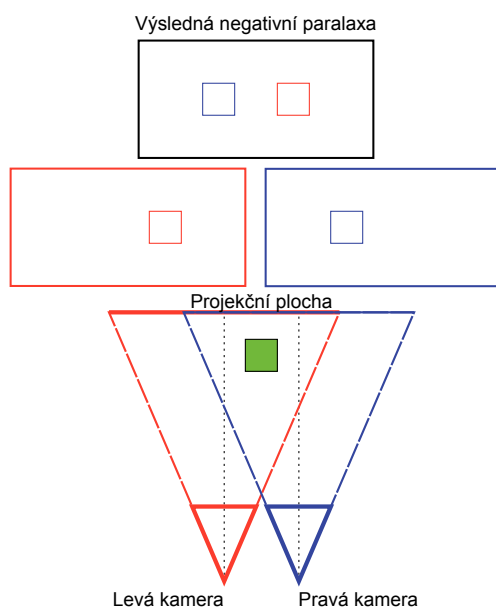
Negativní parallaxa bývá obvykle vhodná v situacích, kdy je potřeba vytvořit scénu, která vystupuje z projekční plochy k divákovi. Negativní parallaxu mohou například využít tvůrci animací, pokud si přejí, aby diváci podvědomě uhýbali létajícím kulkám, explozím nebo částicím v jakékoliv formě. Tuto parallaxu je nutné vhodně využít. Není nejlepší volba ji využívat v situacích, kde bude vystoupilý objekt záhy ořezán rámem projekční plochy. V tu chvíli by vznikl nepříjemný jev kolidující s psychologickými vodítky, které byly popsány v předchozích kapitolách. Vzdálenější rám projekční plochy by totiž v tu chvíli překrýval objekty, které se divákovi jeví jako bližší.



Obrázek 11: Je důležité určit, kdy je vhodná pozitivní a kdy negativní paralaxa[9]

Naopak pozitivní paralaxa dává divákům možnost vychutnat si plnou hloubku scény v pozadí. Při využití této metody je scéna rozprostřena za projekční plochou a nedochází k žádným nepříjemným kolizím objektů s okrajovým rámem displeje. Pokud se stane, že je nějaký objekt scény překrytý rámem projekční plochy, pak se nejedná o nic jiného než překrytí vzdáleného objektu objektem bližším.

Problém, který nastal, byl ten, že při využití paralelního získávání stereopárů se vždy dosáhne úplné negativní paralaxy. Jak je vidět na obrázku níže, nemůže nastat situace, kdy se vykreslí plnohodnotná scéna s pozitivní paralaxou. Obrazy získané paralelní metodou budou směrem k divákovi zvětšovat svou rozteč. Tyto obrazy budou vždy umístěny tak, že obraz pro levé oko se nachází napravo od obrazu pro pravé oko.



Obrázek 12: Získání stereopáru paralelní metodou

Určité řešení poskytla uměle vytvořená „falešná pozitivní paralaxa“. Tato úprava má svůj základ ve funkci pseudoskopu. Zjednodušeně řečeno, oba obrazy s negativní para-

laxou, získané při paralelní metodě, budou prohozeny tak, aby obraz pro levé oko byl umístěn vlevo a obraz pro pravé oko vpravo. Touto výměnou bude zajištěna konvergence očí až za projekční plochou. Jedinou vlastností pozitivní paralaxy, kterou nebylo možné nastolit, byla rozteč obrazů. Rozteč obrazů by se měla správně zvětšovat s rostoucí vzdáleností mezi projekční plochou a daným objektem. V tomto případě byla bohužel zachována vlastnost původní negativní paralaxy. Při testech však nenastala ani jedna situace, kdy by tato netradiční metoda byla pro diváka jakkoliv nepříjemná. Velký podíl na tom, že měla tato metoda úspěch, má pravděpodobně i to, že zde figurovala některá psychologická vodítka, kterými divák určoval hloubku ve scéně. Divák, jakožto tvůrce této scény, rovněž ví, v jaké hloubce se nachází jednotlivé objekty a podvědomě tuto informaci dokáže zpracovat v souladu s touto metodou. Dokonce i pro diváka, který není tvůrcem scény, je k dispozici celá řada psychologických vodítek, která dokáže mozek analyzovat při vyhodnocení hloubky jednotlivých objektů (například překrývání, velikost objektů atd.). Tato umělá pozitivní paralaxa je však spíše experimentální a nikde není zaručeno, jakým způsobem na ni bude divák reagovat.

5.5 Viewport

Nejdůležitější částí této práce byla úprava funkcí, které vykreslují scénu uvnitř viewportu. Tyto vykreslovací funkce se nachází uvnitř souboru *view3d_draw*. V tomto souboru, zároveň před samotným vykreslováním, probíhá nastavení kamery, která snímá scénu. V předchozí části bylo popsáno, jak byla tato kamera upravována. Úprava nastavení kamery se provádí v metodě *view3d_main_area_draw_objects*. Tato metoda vykresluje všechny objekty, které jsou umístěny ve scéně. Pod touto metodou se nachází druhá podobná metoda *view3d_main_area_draw_info*, která vykresluje dodatečné informace jako názvy objektů, scény nebo 3D kurzor. Obě tyto metody jsou pak součástí hlavní vykreslovací metody s názvem *view3d_main_area_draw*.

Vykreslování objektů uvnitř *view3d_main_area_draw_objects* se dá rozdělit do několika kroků:

- **Deklarace proměnných** – Na začátku vykreslovací metody probíhá deklarace a definice proměnných používaných v jejím průběhu.
- **Inicializace** – Inicializace, nebo restart určitých částí jako například shadow bufferu, světla a případně vymazání color a depth bufferu.
- **Nastavení vykreslovací matice** – Z nastavení scény se vypočítá a přenastaví vykreslovací matice, která určuje uživatelský pohled.
- **Vykreslení objektů scény** – Vykreslení všech objektů ve scéně. Mohou se vykreslovat například světla, kamery, modely (označené, neoznačené, v editačním režimu atd.), manipulátory nebo řídicí mřížka.

V první části vykreslovací metody věnované deklaraci proměnných byly přidány nové proměnné, které byly využívány v průběhu této metody pro definici druhu stereoprojekce, rozteče mezi stereopáry nebo indexu právě vykreslovaného oka. Do těchto

proměnných byly uloženy hodnoty, které se nachází uvnitř struktury *bContext*. Tyto hodnoty byly z kontextu získány pomocí nově definovaných funkcí zmiňovaných v předešlé kapitole zabývajících se strukturou *bContext*.

Pro účely této práce nebylo potřeba nijak upravovat druhou inicializační část metody. Veškeré osvětlení nebo buffery se nadále využívají v původním nastavení. Jedinou, pro nás důležitou, věcí nacházející se v této části, bylo mazání *color* a *depth* bufferu. Toto mazání bylo nutné využít i v následující části věnované samotnému vykreslování.

Nastavení vykreslovací matice neprobíhá přímo v této metodě. Je zde pouze volána funkce, která tuto matici upraví v závislosti na nastavení scény. Uvnitř této pomocné funkce se z nastavení scény získá projekční a modelová matice. Obě tyto matice jsou pak po určitých úpravách načteny OpenGL. V jednom z připojených hlavičkových souborů s názvem *view3d_intern* je přidána nová globální proměnná. Tato proměnná slouží jako ukazatel na umístění obou používaných matic. Pomocí této globální proměnné bylo možné měnit matice tak, aby bylo dosaženo požadovaného posunutí pro jednotlivé objekty ve scéně.

Je-li potřeba získat dva od sebe odsazené obrazy, je nutné požadované objekty vykreslit dvakrát, pokaždé s odlišným nastavením kamery. Tohoto dvojnásobného překreslení každého viditelného objektu uvnitř scény je docíleno přidáním cyklu o dvou iteracích. Pokud je stereomód aktivní, pak se v tomto cyklu při prvním průchodu nastavuje kamera pro levé oko a při druhém cyklu pro pravé oko. Zároveň jsou v závislosti na použité metodě prováděny určité úpravy vykreslované scény tak, aby vlastnosti metody odpovídaly právě vykreslovanému oku. Pokud je stereomód deaktivován, pak se cyklus provede vždy jen jednou bez stereoskopických úprav.

Před vykreslováním scény probíhá také kontrola objektů, která má za úkol zajistit případnou aktualizaci nastavení stereoprojekce definované pomocí přidaného GUI. Tvorba a funkce samotného GUI je popsána v kapitole 6. Po této kontrole je zde implementováno získávání ohraničení vykreslované scény, které se pro potřeby stereokurzu ukládá do seznamu uvnitř třídy *ms*.

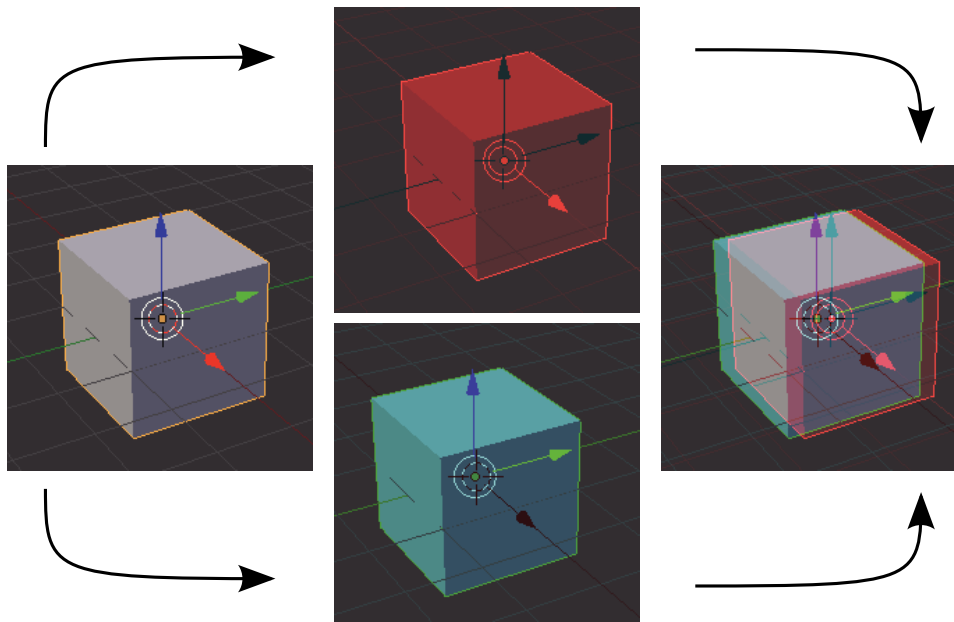
V následující části budou popsány jednotlivé stereoskopické metody, které byly v této práci využity. Bylo důležité pokrýt takové stereoskopické metody, které bude možné využít v praxi. Jsou například úplně vypuštěny metody *above-below* a *side-by-side*. Důvod odstranění těchto metod byl ten, že jsou převážně používány ve stereoskopických přehrávačích, nebo na specializovaných projekčních zařízeních.

5.5.1 Anaglyph

Jak bylo řečeno v kapitole 4.1.1, metoda anaglyph funguje na principu rozlišování stereopárů pomocí barev. Bylo tedy nutné vytvořit barevnou masku, která by dané snímky oddělila. Pro podobné případy lze využít OpenGL příkaz *glColorMask*. Pomocí tohoto příkazu je možné aktivovat a deaktivovat zápis jednotlivých barevnostních kanálů do vykreslovacího bufferu. Parametry, které tento příkaz přijímá, jsou typu *GLboolean* a reprezentují složky RGBA. Nastavením parametrů na hodnotu *GL_TRUE* nebo *GL_FALSE* je možné OpenGL říci, které barevné složky vypustit a které zachovat. Původní inicializační parametry tohoto příkazu jsou všechny standardně nastaveny na *GL_TRUE*.

Důvody pro implementaci této metody byly její jednoduchost a nízká provozní cena. Prakticky každý uživatel programu Blender se stereoskopickou úpravou má možnost si tuto metodu vyzkoušet bez nutnosti pořizování drahého hardwaru. Samotná implementace pak probíhala následovně:

- Při prvním průchodu cyklem probíhá nastavení kamery pro levé oko. Je zavolán příkaz `glColorMask(GL_TRUE, GL_FALSE, GL_FALSE, GL_TRUE)`, kterým se určí, že vykreslované barvy budou v odstínech červené.
- Jsou vykresleny objekty ve scéně.
- Při druhém průchodu cyklem probíhá nastavení kamery pro pravé oko. Před zavoláním příkazu `glColorMask` se provede vymazání hloubkového bufferu tak, aby se oba vykreslené obrazy prolnuly.
- Po vymazání hloubkového bufferu proběhne nastavení barevné masky příkazem `glColorMask(GL_FALSE, GL_TRUE, GL_TRUE, GL_TRUE)`. Tím je zajištěno, že vykreslované barvy budou v odstínech azurové.
- Znova jsou vykresleny objekty ve scéně.
- Po ukončení cyklu se vrací nastavení kamery i barevné masky do normálu, tak aby nebylo ovlivněno další vykreslování.



Obrázek 13: Tvorba stereoprojekce metodou anaglyph

Tato metoda má naneštěstí i své nevýhody. Nejčastějším problémem je částečná nebo i kompletní ztráta barev, což v modelování samotném nemusí být problém. Bohužel při komplexní práci s grafikou je ztráta barev nemyslitelná. Pokud navíc uživatel pracuje s objekty, které jsou například v odstínech červené, modré nebo azurové, vzniká nepříjemné namáhání očí. Zároveň mohou v závislosti na použitém hardwaru nebo kvalitě brýlí vznikat tzv. „duchové“. Ghosting samotný byl popsán v kapitole 3.2. Při testování byla tato metoda přijata s největší negativní odezvou. Nejlepší využití této metody bylo zjištěno při modelování objektů zobrazovaných metodou wireframe (síťový model).

5.5.2 Vertical interlace

Metoda sloupcového prokládání funguje na principu prolnutí dvou obrazů v jeden s tím, že jednotlivé sudé sloupce jsou zobrazovány pro jedno oko, zatímco liché sloupce pro oko druhé. Bylo tedy nutné vytvořit takové prokládací masky, které by při vykreslování propouštěly vždy pouze daný druh sloupců. Vytvoření takovéto masky nebylo možné provést jediným příkazem, jako tomu bylo u metody anaglyph. Bylo zapotřebí zcela jiného přístupu.

Přes celou plochu viewportu, ve kterém se scéna vykresluje, byl natažen obdélník. Na vykreslování tohoto obdélníku byl aplikován OpenGL příkaz *glPolygonStipple*. Tento příkaz udává vykreslovaným polygonům určitý opakující se vzorec. Využitím tohoto příkazu a v něm použité masce je možné definovat vzor vykreslovaného obdélníku. V tomto konkrétním případě se jedná o svisle pruhovaný vzor, kde se jednotlivé pruhy střídají ob jeden pixel. Tímto byla definována maska pro jedno oko. Při vykreslování objektů pro druhé oko je na stejném principu využita maska s podobným, avšak o jeden pixel posunutým vzorem.

Na takto vytvořenou masku byl poté aplikován známý příkaz *glColorMask(GL_FALSE, GL_FALSE, GL_FALSE, GL_FALSE)*. Tím bylo zajištěno, že daná svisle pruhovaná maska bude pro uživatele neviditelná. Díky vlastnostem OpenGL bylo nyní docíleno toho, že pixely vykreslovaných objektů, které byly překryty neviditelnými pruhy budou rovněž skryty[36]. Za tento jev může hloubkový buffer, který ořezává vykreslované polygony. OpenGL optimalizuje vykreslování tím způsobem, že části polygonů, které jsou překryty jinými objekty, nebudou vykresleny. Tato funkce funguje i v případě, že jednotlivé svislé pruhy, které stíní vykreslovanou scénu, mají alfa kanál snížený na nulu. Tím je zajištěno, že se daná scéna vykresluje kompletně svisle pruhovaně, ať se jedná o polygony, linky nebo samotné vertexy.

Stejným způsobem byly v druhé iteraci vytvořeného cyklu vykresleny objekty pro druhé oko. Rozdíl mezi první a druhou iterací je ten, že je využito jiného nastavení prokládací masky (posun o jeden pixel). Takto bylo docíleno vzájemného proložení dvou obrazů stereopáru.

Důvod pro vytvoření této prokládané metody byl ten, že je možné ji využít spolu s autostereoskopickým displejem, případně u některých druhů pasivní projekce s polarizačními filtry. Implementace samotného vykreslování je v tomto případě založena na podobném principu jako u metody anaglyph. Rozdíly jsou jen ty, že při iteraci cyklem není upravována maska barev, ale vzorů. Celkové vykreslování probíhá následovně:

- Při prvním průchodu cyklem probíhá nastavení kamery pro levé oko.
- Je vyvolán příkaz *glPolygonStipple* spolu s parametrem, kterým určíme, že právě používaná maska je ta pro levé oko.
- Přes viewport je vykreslen obdélník, na který je aplikován předdefinovaný vzor a příkazem *glColorMask* deaktivována jeho viditelnost.
- Po vykreslení obdélníku je příkaz *glPolygonStipple* deaktivován tak, aby nedošlo k ovlivnění dalšího vykreslování objektů ve scéně. Zároveň je příkaz *glColorMask* uveden do základního stavu.
- Jsou vykresleny objekty ve scéně.
- Druhý průchod cyklem pak probíhá obdobně. Rozdíl je v použité masce pro příkaz *glPolygonStipple*.

```
for(i = 0; i < 32; i++)
{
    left_vinterlace_mask[i] = 0x55555555;
    right_vinterlace_mask[i] = 0xAAAAAAAA;
}
```

Výpis 4: Nastavení vertikálních prokládacích masek

Výše uvedené masky se pak v závislosti na vykreslovaném oku volaly jako parametr uvnitř příkazu *glPolygonStipple*. Pokud byla scéna vykreslována například pro levé oko, pak byl tento příkaz ve tvaru *glPolygonStipple(left_vinterlace_mask)*. Samotné prokládací masky jsou uloženy uvnitř kontextové struktury *ms*. Inicializace těchto masek probíhá při spuštění programu Blender uvnitř přidáné metody *eye_set*.

Výhoda této metody je bezesporu ta, že zde nedochází ke zkreslení vykreslovaných barev. Při použití autostereoskopického displeje lze za výhodu navíc považovat i to, že není zapotřebí žádných brýlí, které by některým lidem mohly překážet. Bohužel, stejně jako při jiných metodách prokládaného vykreslování, dochází ke snížení rozlišení. V tomto případě je daná scéna vykreslována s polovičním rozlišením na šířku oproti normálu. Rozlišení na výšku zůstává nezměněno.

5.5.3 Horizontal interlace

Metoda řádkového prokládání funguje podobně metoda sloupcového prokládání v kapitole 5.5.2. V tomto případě je však jinak nastavena maska pro vykreslování. Rozdíl je tedy v tom, že namísto střídajících se sloupců, ve kterých jsou promítány obrazy pro pravé a levé oko, se dané vykreslování provádí do vzájemně prokládaných řádků.

Důvody pro implementaci této metody byly tytéž jako u předchozí metody. Autostereoskopické displeje, případně projekce s polarizačními filtry mohou být nastaveny jak pro horizontální, tak pro vertikální prokládání. Samotné vykreslování pak probíhá stejně, jako tomu bylo v předchozím případě.

```

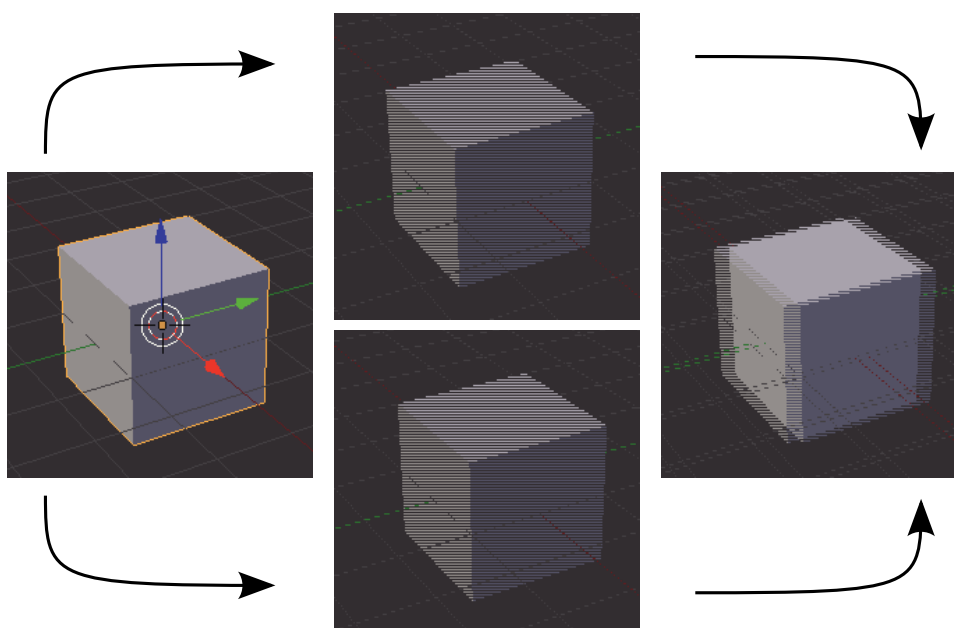
for(i = 0; i < 32; i++)
{
    hinterlace_mask[i] = (i&1)*0xFFFFFFFF;
}

hinterlace_mask[32] = 0;

```

Výpis 5: Nastavení horizontální prokládací masky

Z výše uvedené ukázky zdrojového kódu je vidět, že oproti vertikálnímu prokládání je použita pouze jedna maska, která se dá využít pro obě oči. V závislosti na vykreslovaném oku je pak pouze určen posun této masky. Pokud například hodláme vykreslovat scénu pro pravé oko, má výsledný příkaz formát *glPolygonStipple(&hinterlace_mask[1])*.



Obrázek 14: Tvorba stereoprojekce prokládanou metodou

Stejně zůstávají výhody i nevýhody této projekce. Rozdíl je opět pouze ten, že ke ztrátě plného rozlišení dochází ve výšce vykreslovaného obrazu. Rozlišení na šířku tedy zůstává nezměněno.

5.5.4 Quad buffer

Tato metoda je v dnešní době jedna z nejrozšířenějších. Jak bylo popsáno v teoretické části věnované aktivní stereoprojekci, je zde vhodné zvýšit vykreslovací kmitočet tak, aby bylo možné danou scénu vykreslovat pro každé oko v rozumné frekvenci. Tohoto zvýšení rychlosti vykreslování a jeho rozdělení pro dané oči je docíleno využitím grafických karet s podporou quad bufferu. Zároveň je nutná podpora případných vyšších

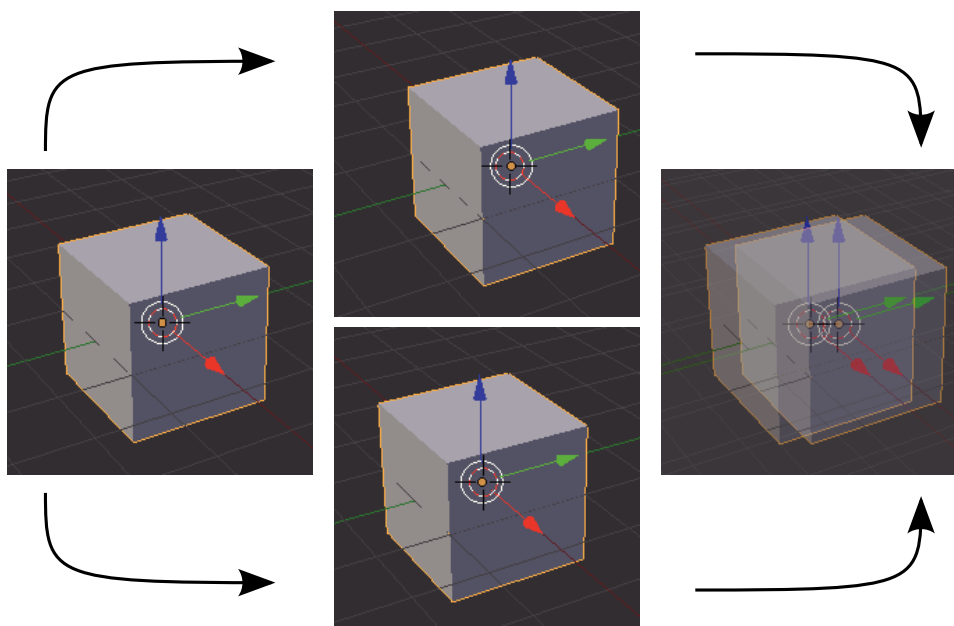
frekvencí u použitého projekčního zařízení. Největší výhody této metody jsou bezesporu ty, že zde nedochází ke zkreslení vykreslovaných barev ani ztrátě rozlišení. Quad buffer, který obvykle nebývá zakomponován v běžném typu grafických karet, lze dnes i emulovat pomocí volně dostupných programů.

Běžné grafické karty využívají pro vykreslování scény double buffer. Při práci s obecnou OpenGL scénou mnohdy dochází k určitým transformacím její kamery a objektů, které obsahuje (rotace, posuny atd.). Při každé takové změně je nutné danou scénu aktualizovat a překreslit tak, aby byly tyto úpravy viditelné. Překreslení scény zpravidla probíhá tak, že je veškerý zobrazovaný obsah vymazán a znovu vykreslen. Samotné vykreslení však může nějakou dobu trvat. Díky této prodlevě pak může uživatel vnímat určité problikávání. Tomuto nepříjemnému efektu je zabráněno právě využitím dvou bufferů (double bufferu). Do jednoho z těchto bufferů se daná scéna vykresluje (back buffer) a jeden je promítán uživateli (front buffer). Když je pak potřeba scénu překreslit, jsou tyto dva buffery pouze prohozeny[32].

Dalo by se říci, že quad buffer je složen ze dvou double bufferů najednou. Tyto dva buffery jsou rozděleny na levý a pravý s tím, že je pak každý přiřazen jednomu oku. Původně tedy byla daná scéna zapisována do zadního bufferu pomocí příkazu *glDrawBuffer(GL_BACK)*. Aby bylo možné quad buffer využít, bylo potřeba jej aktivovat. Tato aktivace probíhala v souboru *wm_window* pomocí funkce *wm_window.add_ghostwindow*. Tato funkce vytváří nové GHOST okno. Uvnitř této funkce bylo potřeba při definici nového GHOST okna nastavit příznak, který zajišťoval inicializaci quad bufferu (pokud je přítomen). S využitím quad bufferu se scéna vykresluje následujícím způsobem:

- Při prvním průchodu cyklem probíhá nastavení kamery pro levé oko. Příkazem *glDrawBuffer(GL_BACK_LEFT)* se pro zápis připraví levý zadní buffer.
- Jsou vykresleny objekty ve scéně.
- Při druhém průchodu cyklem probíhá nastavení kamery pro pravé oko. Příkazem *glDrawBuffer(GL_BACK_RIGHT)* se pro zápis připraví pravý zadní buffer.
- Opět jsou vykresleny všechny objekty ve scéně.
- Po ukončení cyklu se vrací nastavení kamery i bufferu do normálu tak, aby nebylo ovlivněno další vykreslování. Další zápis do zadního bufferu je nastaven na *glDrawBuffer(GL_BACK)*. Toto nastavení zaručí, že další vykreslování bude prováděno do obou zadních bufferů.

Nepříjemné situace mohou nastat například u citlivých jedinců, kteří mohou vnímat občasně blikání, které bylo popsáno v teoretické části věnované aktivní projekci. Toto blikání může být zároveň způsobeno nekvalitním hardwarem, nebo i špatnými okolními podmínkami. Nevýhodou může být i vyšší pořizovací cena vybavení (aktivní brýle, kompatibilní projekční zařízení a grafická karta).



Obrázek 15: Tvorba stereoprojekce metodou quad buffer

5.6 Úprava klávesových zkratk

Pro snazší nastavování stereoskopické rozteče byly vytvořeny určité klávesové zkratky, které měly za cíl uživateli zjednodušit ovládání. Blender samotný využívá nespočetné množství klávesových zkratk, které jsou navíc závislé i na momentálním stavu aplikace (jsou zde například rozdílné funkce některých zkratk v editačním módu a mimo něj). Důležité faktory tedy byly takové, aby byla klávesová zkratka nevyužitá a zároveň smysluplná.

Podobně jako u registrace parametrů před spuštěním, které probíhaly uvnitř souboru *creator*, tak i zde je potřeba registrovat klávesové zkratky a jimi vyvolané metody. Samotná definice a registrace nově definovaných klávesových zkratk a jejich metod je umístěna v souboru *screen.ops*. Blender obsahuje i jiné podobné soubory, uvnitř kterých probíhají definice dalších zkratk a metod. Tyto soubory se pak zpracovávají v závislosti na stavu aplikace. Postup tvorby klávesových zkratk je následující:

- **Registrace zkratk** – Je nutné zaregistrovat požadovanou zkratku a určit klávesy, kterými bude vyvolána daná metoda.
- **Registrace metody** – Do seznamu procházených metod je také nutné přidat název nové předdefinované metody.
- **Definice metody** – Po jejím zaregistrování je nutné danou metodu definovat. Uvnitř této metody je pak přiřazena akce, která bude provedena.

- **Definice akce** – Tato funkce reprezentuje akci, která bude po stisku dané klávesové zkratky vyvolána.

```

static int mystereo_up(bContext *C, wmOperator *UNUSED(op), wmEvent *event)
{
    float distance = getEyeDistance(C);
    float state = getStereoState(C);

    if (state > 0)
        distance = distance + 0.05;

    setEyeDistance(C,distance);

    return OPERATOR_FINISHED;
}
...
static void SCREEN_OT_stereo_up(struct wmOperatorType *ot)
{
    ot->name = "Stereo Spacing Up";
    ot->description = "Stereo spacing up";
    ot->idname = "SCREEN_OT_stereo_up";

    ot->invoke= mystereo_up;
    ot->poll= ED_operator_screenactive;
}
...
WM_operatortype_append(SCREEN_OT_stereo_up);
...
WM_keymap_add_item(keymap,"SCREEN_OT_stereo_up",PADPLUSKEY,KM_PRESS,KM_CTRL,0);

```

Výpis 6: Registrace klávesových zkratk a definice jejich funkcí

Na výše uvedeném zdrojovém kódu je vidět definovaná funkce, která se provede vždy při stisku dané klávesové zkratky. Pod funkcí je umístěna obecná předdefinovaná metoda, uvnitř které je tato funkce registrována. V další části kódu je pak umístěna registrace předdefinované metody. Na konci výpisu zdrojového kódu je definice požadované klávesové zkratky a přiřazení názvu předdefinované metody, kterou tato zkratka bude při aktivaci vyvolávat. V tomto případě se jedná o zvětšení stereoskopické rozteče při stisknutí kláves *ctrl* + *plus*.

6 Python

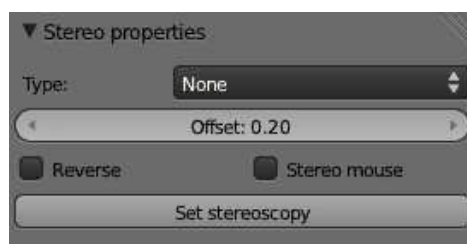
Python je dynamický objektově orientovaný skriptovací programovací jazyk, který v roce 1991 navrhl Guido van Rossum[22]. Tento jazyk bývá již delší dobu využíván pro tvorbu GUI, skriptů a mnoha uživateli definovaných funkcí programu Blender. Samotný Blender má pro tento jazyk vytvořeno vlastní rozhraní, skrze které pak komunikuje s danými skripty.

Z důvodů větší uživatelské přívětivosti bylo pomocí jazyku Python vytvořeno grafické rozhraní. Skrze toto rozhraní může uživatel spravovat všechna stereoskopická nastavení, která tato verze programu Blender nabízí. Je důležité poznamenat, že stereoskopická projekce jako taková ke své funkčnosti toto rozhraní nepotřebuje. GUI je vytvořeno jen pro usnadnění práce se stereoskopickými nastaveními. Bez rozhraní je možné tato nastavení definovat pomocí klávesových zkratk a parametrů při spuštění. Samotné GUI je pak možné v programu spustit dvěma způsoby:

- **Addon** – GUI je možné nainstalovat podobně jako klasický addon. Instalace se nachází v menu *User Preferences...* (ctrl + alt + u), v záložce *Addons*. Zde pak stačí zvolit možnost *Install Addon...* a vyhledat požadovaný soubor se skriptem. Po této instalaci je pak nutné daný addon aktivovat. Aktivace se provede vybráním přídatku ze seznamu. Addon reprezentující GUI pro ovládání stereoskopie se nazývá „*Stereoscopic properties GUI*“.
- **Skript** – Další možností instalace tohoto skriptu je uvnitř textového editoru v programu Blender. Po spuštění editoru *Text Editor* je nutné vytvořit nový soubor. Pro tvorbu nového souboru slouží tlačítko *New*. Když je pak takový soubor vytvořen, stačí do něj vložit obsah souboru s daným skriptem. Pro aktivaci skriptu je pak nutné stisknout tlačítko *Run Script*.

Nainstalované uživatelské rozhraní je pak možné nalézt v záložce *Scene*. Dbalo se na to, aby toto rozhraní bylo co možná nejprehlednější a nejjednodušší na ovládání. Grafické rozhraní pak obsahuje následující ovládací prvky:

- **Type** – Výběr určitého typu využívané stereoprojekční metody.
- **Offset** – Nastavení rozteče mezi dvěma snímky stereopáru.
- **Reverse** – Možnost prohození snímků stereopáru. Vytvoří určitou „falešnou“ stereoprojekci s pozitivní paralaxou (princip pseudoskopu).
- **Stereo mouse** – Možnost aktivovat stereokurzor (viz. kapitola 5.3.1).
- **Set stereoscopy** – Tlačítko, kterým je nutné potvrdit nová nastavení stereoskopické projekce.



Obrázek 16: Uživatelské rozhraní

6.1 Skript

Příložený soubor s názvem „stereoGUI.py“ obsahuje skript v jazyce Python, který definuje celé uživatelské rozhraní a jeho funkce. Tento skript by se dal rozdělit na několik částí:

- **Info** – Na začátku souboru se nachází hlavička, která reprezentuje různé informace o daném addonu.
- **Import** – Je důležité importovat balíčky programu Blender, a stejně tak i databázi tříd a proměnných, které lze využít.
- **Hlavička** – Definice, kde se vytvářené GUI bude nacházet.
- **Deklarace prvků** – V této části jsou vytvářeny všechny prvky, které se v GUI objevují. Prvkům se definují jejich vlastnosti, případně definiční obory.
- **Vykreslení prvků** – Část, ve které se všechny definované prvky vykreslují. Určuje se pozice, na které se daný prvek vykreslí.
- **Správa událostí** – Finální část skriptu, ve které se spravují a zpracovávají všechny události a proměnné. Probíhají zde výpočty a potřebné úpravy scény, které je nutno provést před předáním výsledných hodnot dál.

Poslední část skriptu je stěžejní. V této části jsou zpracovány veškeré úpravy, které uživatel při práci s rozhraním provedl. Zároveň se zde vytváří objekt, do kterého se ukládají přenášené informace. Tento objekt je pak „odchycen“ uvnitř zdrojového kódu při vykreslování scény. Před vykreslením samotné scény jsou z tohoto datového objektu vyextrahována a zpracována všechna důležitá data. Po zpracování těchto informací je upraven současný stav stereoprojekce na uživatelem zvolená nastavení. Poté, co jsou provedeny požadované úpravy, je tento datový objekt smazán.

7 Testování

Důležitou součástí této diplomové práce bylo otestování vytvořené odnože programu Blender na dobrovolnících. Tito lidé pocházeli z řad studentů, profesorů i uchazečů o studium na vysoké škole. V průběhu testování bylo získáno čtyřicet šest dotazníků. Testovaní lidé měli s daným programem zkušenosti na různých úrovních. Byli začvičení a testováni nováčci, pokročilí studenti, kteří se grafice již nějaký čas věnují, i pár jedinců na vysoké úrovni. Mezi testovanými subjekty byli muži i ženy.

Každý testovaný subjekt měl možnost si vyzkoušet v dané aplikaci modelovat, animovat, případně, pokud nebyl v této oblasti příliš zdatný, projít virtuální prohlídkou některé předpřipravené scény. V průběhu testování bylo dbáno na to, aby každý z testovaných subjektů měl podobně nastavené okolní podmínky. Okolními podmínkami je myšlena vzdálenost subjektu od projekční plochy, intenzita okolního osvětlení nebo také využívaný hardware. Všem testovaným subjektům byl dán čas na to, aby se s touto úpravou programu Blender sžili a měli možnost si vyzkoušet vše, co uznali za vhodné.

Všem otestovaným lidem byl v závěru předán formulář (viz. obrázek B v příloze) k vyplnění. V průběhu testu byla přítomna i kontaktní osoba, které mohli lidé pokládat případné dotazy. Přítomnost této osoby byla rovněž pozorovacího charakteru. Mnozí dotazovaní své poznatky a nápady pouze pronesli, ale málokdy zapsali nebo vysvětlili. Z tohoto důvodu byla přítomnost pozorovatele nezbytná. Je důležité poznamenat, že pozorovatel nijak neovlivňoval dotazované a upozorňoval také na to, aby všechny subjekty vyplňovaly dotazník pravdivě bez ohledu na možné zábrany v negativní kritičnosti.

7.1 Prvotní testy

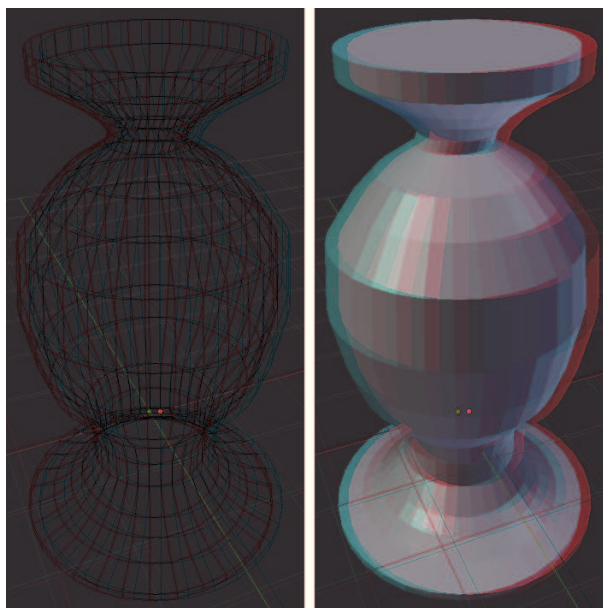
Při prvních testech byli lidé pro porovnání testování jak na aktivní stereoprojekci, tak na stereoprojekci s využitím metody anaglyph. Výsledky těchto testů byly zpracovány a byly zjištěny zajímavé poznatky. V první řadě bylo jasně prokázáno, že lidé více preferovali aktivní metodu s využitím quad bufferu před metodou anaglyph. Dále byly objeveny problémy týkající se zdvojování kurzoru při zaměřování určitých bodů. Konzultacemi s testovanými subjekty bylo zjištěno, že právě nepříjemné zdvojování kurzoru v obou metodách může za většinu získaných negativních ohlasů. Určité procento testovacích subjektů si rovněž stěžovalo na mírnou únavu očí. Taková únava, případně bolest očí, může být způsobena převážně nezvykem na podobně zobrazovanou scénu nebo také určitými charakteristickými vlastnostmi využitých metod.

V průběhu testů bylo rovněž zjištěno, že se změnou provedenou v programu Blender se lépe sžijí hlavně ti uživatelé, kteří s tímto programem nemají větší zkušenosti. Člověk, který se v tomto programu teprve učí, bývá obvykle víc otevřen novým věcem a možnostem než takový uživatel, který má naučené určité způsoby práce a je zvyklý na klasické nestereoskopické zobrazování scény. Je důležité poznamenat, že i zkušenější uživatelé dokázali ocenit zvýrazněnou hloubku scény. Problém u těchto lidí byl spíše nezvyk na nový způsob vykreslování.

7.1.1 Anaglyph

Při testování projekce pomocí metody anaglyph bylo zjištěno, že nejnepříjemnějším faktorem, který odrazoval většinu dotazovaných, byl ghosting. Mezi další nejčastější faktory, na které si dotazovaní uživatelé stěžovali, bylo zkreslení barev a potemnění obrazovky. Všechny tyto faktory měly za výsledek převážně negativní ohlas u většiny dotazovaných. Místy si lidé stěžovali na bolest očí vzniklou právě jejich zvýšeným namáháním. Zároveň zde vzniká určitá binokulární rivalita díky tomu, že každé oko vidí vše v odstínech jiné barvy. Je důležité si uvědomit, že člověk při práci vnímá i své okolí. Lidem se při použití této metody barvově zkreslí i pohled na klávesnici nebo pozadí za projekční plochou.

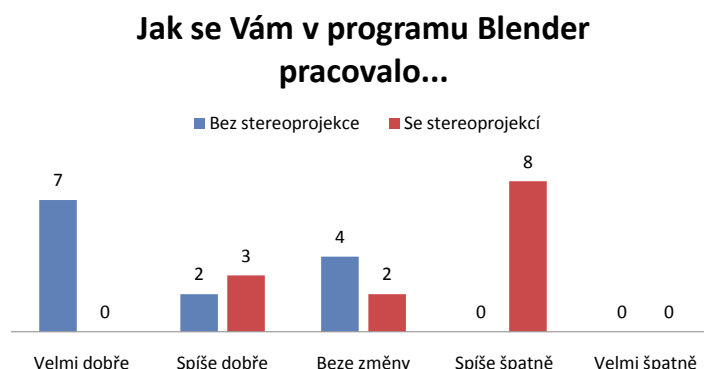
Tato metoda se přes některé své nedostatky setkala také s kladnou odezvou ze strany uživatelů. Díky nim bylo zjištěno, že je možné s touto metodou pracovat bez větších problémů při aktivaci tzv. wireframe módu. Po aktivaci zobrazování pomocí wireframe jsou veškeré objekty ve scéně vykresleny jako síťové modely. Pokud je tedy tento způsob vykreslování aktivní, pak jsou automaticky potlačeny barvy a materiály těchto objektů (síťový model je vykreslen pomocí čar). Jelikož jsou barvy modelů potlačeny, nedochází k přílišnému vzníkání ghostingu nebo barvového zkreslení. Výsledný síťový model je pak bez zvýšené zátěže očí příjemnější na pohled.



Obrázek 17: Rozdíly mezi wireframe (vlevo) a solid (vpravo) zobrazením

Zpracované výsledky dotazníků pro tuto metodu jsou graficky znázorněny níže. Pro srovnání jsou zde znázorněny dojmy z běžného využívání programu Blender bez stereoprojekce proti využívání této aplikace se stereoprojekcí pomocí metody anaglyph. Z uvedeného grafu je patrné, že se metoda anaglyph neseťkala s příliš pozitivní odezvou. Vzhledem ke všem faktorům, které byly v této kapitole popsány, je pochopitelné,

že je tato metoda vhodná pouze pro určitý typ uživatelů. Metodu mohou bezproblémově využívat ti uživatelé, kteří dokáží nastolit takové okolní podmínky, které nevedou ke zbytečné zátěži jejich očí. Těmito podmínkami je myšlena například kalibrace monitoru, využívání wireframe módu při modelování, úprava barev pozadí za projekční plochou nebo také zvolení vhodného osvětlení pracovního prostoru.



Obrázek 18: Výsledky testů metody anaglyph

7.1.2 Quad buffer

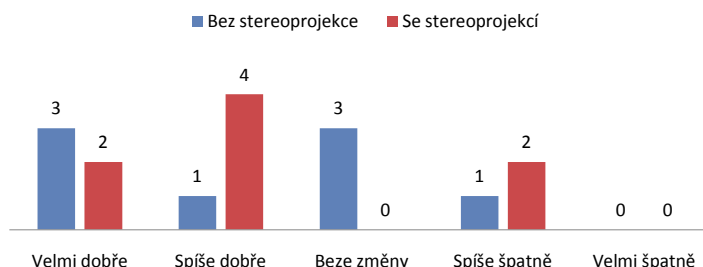
První vlna testování této metody probíhala s využitím projektoru značky Gali-3D. Testování metody quad buffer pomocí monitoru BenQ probíhalo až v pozdějších fázích, ve kterých byly provedeny jisté úpravy dané projekce tak, aby více vyhovovala uživateli. Níže popsaná část tedy odráží testy s využitím projektoru.

Po srovnání výsledků a reakcí uživatelů, kteří vyzkoušeli obě uvedené metody bylo zjištěno, že se metoda s využitím aktivních brýlí setkala s mnohem kladnějším ohlasem než pasivní metoda anaglyph. Využitím této metody nedochází k žádným barvovým zkreslením a není proto potřeba se v práci omezovat pouze na wireframe mód.

Právě při testování této projekce bylo zjištěno, že mnohým uživatelům způsobovalo určité problémy zaměřit kurzorem na požadovaný bod (vertex, linku atd.). Problematika stereokurzu byla popsána v kapitole 5.3.1. Díky rozdvajování kurzoru byly při testech této metody získány určité negativní ohlasy. Další občasné stížnosti se týkaly hlavně zvýšené námahy očí způsobené blikáním brýlí. Uživatelé na blikání reagovali subjektivně. Citlivější jedinci si jsou blikání vědomi a je jim nepříjemné, jiní jej nemusí vnímat vůbec. Blikání samotnému bohužel úplně zamezit nelze, je však možné jej místy potlačit. Výrobci se například snaží zvýšit obnovovací frekvenci projekčních zařízení i brýlí tak, aby bylo blikání méně zřetelné. Mnohdy může záležet i na okolním osvětlení. Například, nachází-li se za projekčním zařízením okno, pak se může blikání jevit zřetelnější.

Zpracované výsledky dotazníků pro tuto metodu jsou graficky znázorněny níže. Pro srovnání jsou zde znázorněny dojmy z běžného využívání programu Blender bez stereoprojekce proti využívání této aplikace se stereoprojekcí pomocí metody quad buffer.

Jak se Vám v programu Blender pracovalo...



Obrázek 19: Výsledky první vlny testů metody quad buffer

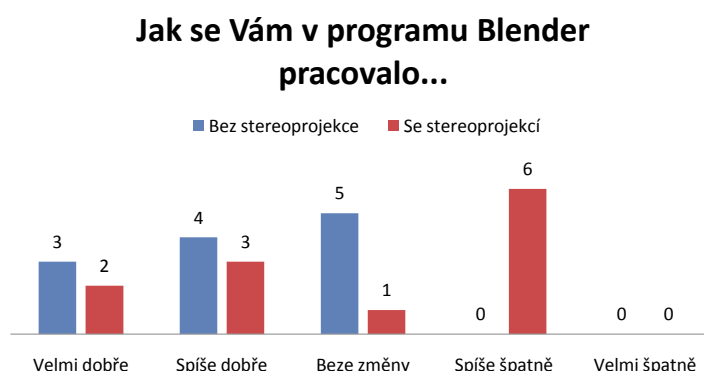
Z grafu lze vyčíst, že tato metoda byla uživateli přijata kladněji než metoda předešlá. Pokud přehlédneme určité minoritní komplikace vzniklé blikáním či rozdvojováním kurzoru, pak by se tato metoda dala označit za přívětivou. Většina uživatelů se vyjádřila kladně k nově získané hloubce scény, která mnohým dopomohla k celkové prostorové představivosti. Mnozí ocenili možnost takto shlédnout různé animace a efekty, které program Blender nabízí. Zmíněné blikání bohužel softwarově úplně potlačit nelze, je proto vhodné věnovat zvýšenou pozornost nákupu kvalitního hardwaru.

7.2 Testy po úpravách

Po zpracování všech dotazníků a reakcí testovaných uživatelů byly na stereoskopické verzi programu Blender provedeny určité změny. Tyto úpravy měly za hlavní cíl reflektovat potřeby a poznatky uživatelů. Za zavedením těchto změn stála určitá snaha o zvýšení uživatelské přívětivosti programu. Mezi úpravy patří:

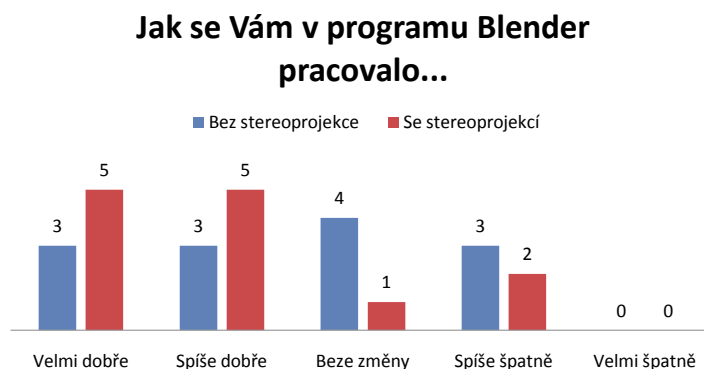
- **Přidání klávesových zkratk** – První změnou bylo přidání klávesových zkratk, kterými si uživatel mohl pohodlněji upravovat stereoskopickou rozteč.
- **Posun přední ořezové roviny** – Při větším přiblížení k objektům docházelo k takovým rozestupům mezi jednotlivými snímky stereopáru, že je mozek nedokázal nadále chápat jako jediný objekt. V těchto případech docházelo ke ghostingu a divergenci očí. Tento problém byl vyřešen posunem přední ořezové roviny. Objekty jsou nyní ořezávány dříve než dochází k „rozpojení“ snímků stereopáru.
- **Vytvoření stereokurzu** – Díky problémům s rozdvojováním klasického kurzoru při zaměřování některých bodů byl vytvořen stereokurzor (viz. kapitola 5.3.1).
- **Přidání ovládacího panelu** – GUI pro ovládání stereoprojekce bylo přidáno až v poslední fázi testování. Jedná se především o kosmetickou úpravu, která má za cíl zvýšení pohodlnosti při výběru a nastavování stereoprojekce.

Zpracované výsledky nových dotazníků pro aktivní metodu quad buffer jsou graficky znázorněny níže. Pro srovnání jsou uvedeny dojmy z běžného využívání programu Blender bez stereoprojekce proti využívání této aplikace se stereoprojekcí.



Obrázek 20: Výsledky testů po úpravě metody quad buffer s využitím projektoru

Výše uvedený graf dokládá mírné snížení kladného přijetí této metody mezi testovanými subjekty. Důvodem tohoto poklesu byly původně nezjištěné chyby v první verzi stereokurzu. Kurzor měl tendence se vykreslovat na jiné pozici, než se opravdu nacházel. Tato odchylka byla způsobena pop-up menu, které uživatel vyvolával. Toto menu při vykreslování negativně ovlivňovalo promítaný kurzor a zkracovalo jeho pravou pozici. Poté, co byly tyto chyby odstraněny, byly provedeny poslední testy. Při této příležitosti byly subjekty zároveň testovány s využitím novějšího hardwaru (aktivní brýle 3D Vision, monitor BenQ).



Obrázek 21: Výsledky testů po úpravě metody quad buffer s využitím monitoru

Poslední graf znázorňuje, že tato stereoprojekce byla přijata ze všech nejlépe. Velkým přínosem bylo odstranění rozdvajování kurzoru, ale také posun přední ořezové roviny. Zde také byla představena poslední verze stereokurzu, která měla odstraněny původní nedostatky. Přidání klávesových zkratk a nového GUI byly spíše kosmetické záležitosti,

kteře měly za úkol pouze zpřijemnit ovládání. Nelze popřít, že za možným zvýšením kvality stojí i využití novějšího hardwaru.

7.3 Hardware

Uvnitř každé srovnávané skupiny (anaglyph, quad buffer s využitím projektoru, quad buffer s využitím monitoru) byl využíván stejný hardware tak, aby nedošlo ke zkreslení jednotlivých dat. Každý v dané skupině testoval stereoskopickou verzi programu Blender za stejných podmínek.

Skupina, která testovala stereoprojekci pomocí pasivní metody anaglyph, byla testována na následně rozepsaném hardwaru:

- **Monitor** – Asus VB195T 19“ s rozlišením 1280x1024
- **Brýle** – Klasické anaglyph brýle červená/azurová značky Prime.

Skupina, která testovala stereoprojekci pomocí aktivní metody s využitím projektoru, byla testována na následně rozepsaném hardwaru:

- **Projektor** – InFocus DepthQ DQ3120-X2 s rozlišením 1024x768
- **Brýle** – 60gx NuVision Gali-3D
- **Grafická karta** – nVidia Quadro FX 3500

Skupina, která testovala stereoprojekci pomocí aktivní metody s využitím monitoru, byla testována na následně rozepsaném hardwaru:

- **Monitor** – BenQ XL241OT 23,6“ s rozlišením 1920x1080
- **Brýle** – nVidia 3D Vision
- **Grafická karta** – nVidia Quadro FX 380

Ghosting, ke kterému docházelo při použití metody anaglyph, byl většinou způsoben rozdílnými odstíny červené a azurové barvy brýlí vůči barvám, které vykresloval monitor. Tyto rozdíly lze potlačit lepší kalibrací monitoru, případně pořízením kvalitních anaglyph brýlí. Kalibrací monitoru je myšlena úprava zobrazovaných barev, kontrastu, sytosti nebo jasů. Úprava těchto vlastností může potlačit, nebo úplně vyřešit ghosting problém při projekci s metodou anaglyph.

V průběhu testů byl místy zaznamenáván i ghosting u aktivní metody s využitím monitoru. Tento jev může být způsoben několika faktory. V první řadě se může jednat o neschopnost monitoru reagovat na rychle měnící se zobrazované snímky. Takový monitor pak není schopen upravit daný změněný pixel v požadovaném čase a tudíž je pak tento pixel ještě částečně viditelný ve snímku vyhrazeném pro druhé oko. Druhým faktorem ovlivňujícím tento nepříjemný jev je teplota monitoru. Pokud není monitor dostatečně zahřátý, pak může být zvýšena i doba mezi překreslením jednotlivých pixelů. V našem

konkrétním případě se jednalo o vliv teploty monitoru, jelikož byl tento jev viditelný pouze při spuštění monitoru s tím, že po určitém čase (řádově pět minut) odezněl.

Dalším hardwarově závislým nedostatkem, se kterým se uživatelé v průběhu testování aktivní metody setkali, byla váha brýlí. Při použití starších brýlí si mnozí uživatelé stěžovali na jejich neforemnost a vyšší hmotnost. Zároveň bylo potřeba, aby tyto brýle byly neustále nasměrovány na synchronizační infračervený emitor. Spojení mezi staršími brýlemi a emitorem navíc nesmělo být ničím překrýváno z důvodu desynchronizace. Tento starší typ brýlí má také tu vlastnost, že výsledný obraz mírně ztmavuje. Takto uživatel nevidí přesně ten jas, který je promítán. Zkreslený jas může být pro uživatele (grafiky především) stejně matoucí a nechťený jev, jako ztráta barevné informace u metody anaglyph.

Novější typ brýlí, který byl použit při testech aktivní projekce s využitím monitoru, byl přijat mnohem lépe. Hmotnost těchto brýlí byla nižší než u brýlí předcházejících. Při testech s těmito brýlemi si žádný z uživatelů nestěžoval na nedostatky spojené s hmotností nebo tvarem. Další výhodou těchto novějších typů brýlí je zvýšení výsledného jasů. Některé typy modernějších brýlí se snaží upravovat výsledný jas tak, aby byl zkreslený co možná nejméně (např. brýle nVidia využívající metodu LightBoost). Další důležitou vlastností některých nových typů brýlí je synchronizace. V případě testů s monitorem byly využity dva druhy synchronizace. Prvním druhem synchronizace bylo, podobně jako u projektoru, využití infračerveného emitru. Výhoda tohoto novějšího emitru je, že není nutný neustálý nepřerušovaný kontakt s brýlemi. Emitor pouze v určitých intervalech provádí synchronizaci s brýlemi, které se pak snaží udržovat nastavenou frekvenci. Tímto je zajištěno, že pokud nastane situace, kdy je emitor blokován, nedojde k okamžité desynchronizaci. Jako druhý způsob synchronizace bylo USB spojení. Pokud byly využity aktivní brýle ve spojení s počítačem pomocí USB, pak nebylo potřeba žádné bezdrátové technologie a tudíž nemohlo dojít k jejich desynchronizaci blokadí emitru. Nevýhodou této metody pak byla maximální možná vzdálenost brýlí od počítače v závislosti na délce USB kabelu.

Nevýhodou využití aktivní projekce za pomoci projektoru bylo osvětlení. Z důvodu co nejkvalitnější projekce zde byla snaha snížit osvětlení na nejnižší možnou úroveň. Pokud bylo osvětlení pracovní místnosti příliš jasné, pak byla projekce na plátno nekvalitní. Při jasném světle zároveň místy docházelo k problémům se synchronizací mezi použitými brýlemi a starším infračerveným emitrem. Někteří testovaní uživatelé měli vzhledem ke svým dosavadním zkušenostem občasnou potřebu sledovat klávesnici, což bylo s využitím tmavších aktivních brýlí a sníženém osvětlení v místnosti náročné. Žádné z těchto problémů se nevyskytovaly s využitím monitoru, novějšího typu brýlí a kvalitnějšího emitru. V případě USB spojení mezi brýlemi a počítačem se není navíc důvod obávat o kvalitu synchronizace v závislosti na osvětlení.

Je důležité poznamenat, že poznatky obsažené v této kapitole nemají nijak ovlivnit nebo zvýhodnit určitého výrobce výše popsaného hardwaru. Prováděné testy neměly za účel dokázat možné vzájemné rozdíly nebo případné nedostatky mezi jednotlivými projekčními zařízeními a použitými brýlemi. Jedná se pouze o informativní kapitolu, která je zaměřena na popis využitých přístrojů.

8 Závěr

Cílem této diplomové práce bylo nastudování rozsáhlé problematiky týkající se stereoprojekce, dále pak zjištění možností její implementace v grafickém open source programu Blender a na závěr otestování výsledných úprav na vzorku uživatelů. Při studiu samotné stereoprojekce jsem nenarazil na žádné větší problémy. Zjistil jsem, jaké jsou v dnešní době využívány způsoby získávání stereopárů, nebo následné možnosti projekce těchto obrazů uživatelům. Po zpracování této problematiky byla pro implementaci vybrána paralelní metoda získávání stereopárů. V závislosti na této vybrané metodě bylo při promítání docíleno úplné negativní paralaxy. Pro stereoprojekci byla zvolena metoda anaglyph, prokládaná metoda a metoda s využitím quad bufferu.

Před samotnou implementací stereoskopického rozšíření byl program Blender zanalyzován a prozkoumán. Byla zmapována část, která se zabývá vykreslováním scény a spolu s ní i další části, které jsou s vykreslováním úzce spjaté. Objevil jsem možnosti struktur, které Blender nabízí a využil jsem stávající architekturu a složení programu pro účely přidané stereoprojekce. Do zdrojového kódu bylo nutné přidat některé vlastní proměnné, struktury a metody, které měly za úkol pomoci v nově upraveném vykreslování nebo průběžných výpočtech. Implementace pak probíhala tím způsobem, že byl ve vykreslovací části vytvořen cyklus, který měl za úkol překreslovat danou scénu pro obě oči. Uvnitř tohoto cyklu pak probíhaly požadované úpravy týkající se nastavení kamery nebo druhu stereoprojekce. Při spuštění programu byla vytvořena inicializace stereoprojekce a jejích parametrů. Ve zmapovaném programu pak bylo nutné vytvořit určitý nosič různých stereoskopických nastavení a informací. Jako nosič byla využita stávající kontextová struktura, kterou Blender samotný využíval k přenosu svých důležitých dat skrze celou aplikaci. V průběhu práce s tímto programem bylo nutné vytvořit přijatelný způsob ovládání provedených úprav. Díky této potřebě vznikly určité klávesové zkratky, spouštějící parametry nebo dokonce grafické rozhraní.

V průběžných uživatelských testech byly zjištěny určité nedostatky v některých zpracovaných stereoprojekčních metodách. Některé z těchto nedostatků byly odstraněny, jiné se podařilo z větší části potlačit. Docházelo například k rozdvajování kurzoru, které bylo vyřešeno tak, že byl tento kurzor vykreslován pouze pro jedno oko. Jeden z dalších objevených nedostatků byl ten, že při větším přiblížení k objektu mohlo docházet k roztrhnutí obrazů jeho stereopáru. Toto roztrhnutí bylo potlačeno tím, že byla posunuta přední ořezová rovina scény. Tento posun měl za následek ořezávání stereoskopických obrazů ještě před tím, než došlo k jejich roztrhnutí. Po zpracování nejčastěji se vyskytujících problémů byla nová verze programu prezentována další skupině testovacích subjektů.

Ve finále bylo získáno čtyřicet šest dotazníků. Po zpracování všech dotazníků a uživatelských ohlasů bylo zjištěno, že nejlépe přijatou technikou byla metoda s využitím quad bufferu. Uživatelé na tuto stereoprojekci reagovali většinou kladně. Místy docházelo ke stížnostem na nepříjemné blikání nebo zvýšenou námahu očí. Téměř všichni uživatelé zaznamenali rozdíl ve vnímané hloubce scény a ocenili nově nabytou perspektivu.

Ukazuje se, že s využitím stereoprojekce se v počítačové grafice a konkrétně v modelování otevírají nové možnosti použití. S využitím stereo zobrazení při návrhu a modelování ve 3D nástrojích mají nyní uživatelé a umělci možnost vidět modelovanou scénu

v doslova novém rozměru. Při testech bylo rovněž ukázáno, že stereoskopie jako taková nemusí být využívána pouze pro projekční účely, ale také jako pomoc grafickým designérům při jejich tvorbě.

Další možností, jak na tuto práci navázat a vyřešit některé problémy týkající se stereoskopie obecně, by mohlo být vylepšení stereokurzoru nebo například zakomponování dalších projekčních metod. Tato práce a data z ní získaná by také mohly sloužit jako impuls pro nově otevíranou sofistikovanější oblast holografie a jejího využití. I když existuje určité procento lidí, kteří nemají rozvinuté prostorové vnímání, čímž výsledné 3D stereo zobrazení nevyužijí, pro ostatní je tato projekce velice zajímavým rozšířením a možností nejen pro 3D modelování. Nově rozšířený Blender navíc ukázal, že práce a modelování s využitím stereoskopie je nejen možná, ale pro mnohé uživatele i poměrně snadná.

Výsledky této práce byly shrnuty v článku s názvem „3D Modeling Using Stereo Projection“, který byl zaslán k přijetí na konferenci WSEAS, která se koná v červenci 2012 na řeckém ostrově Kos.

9 Reference

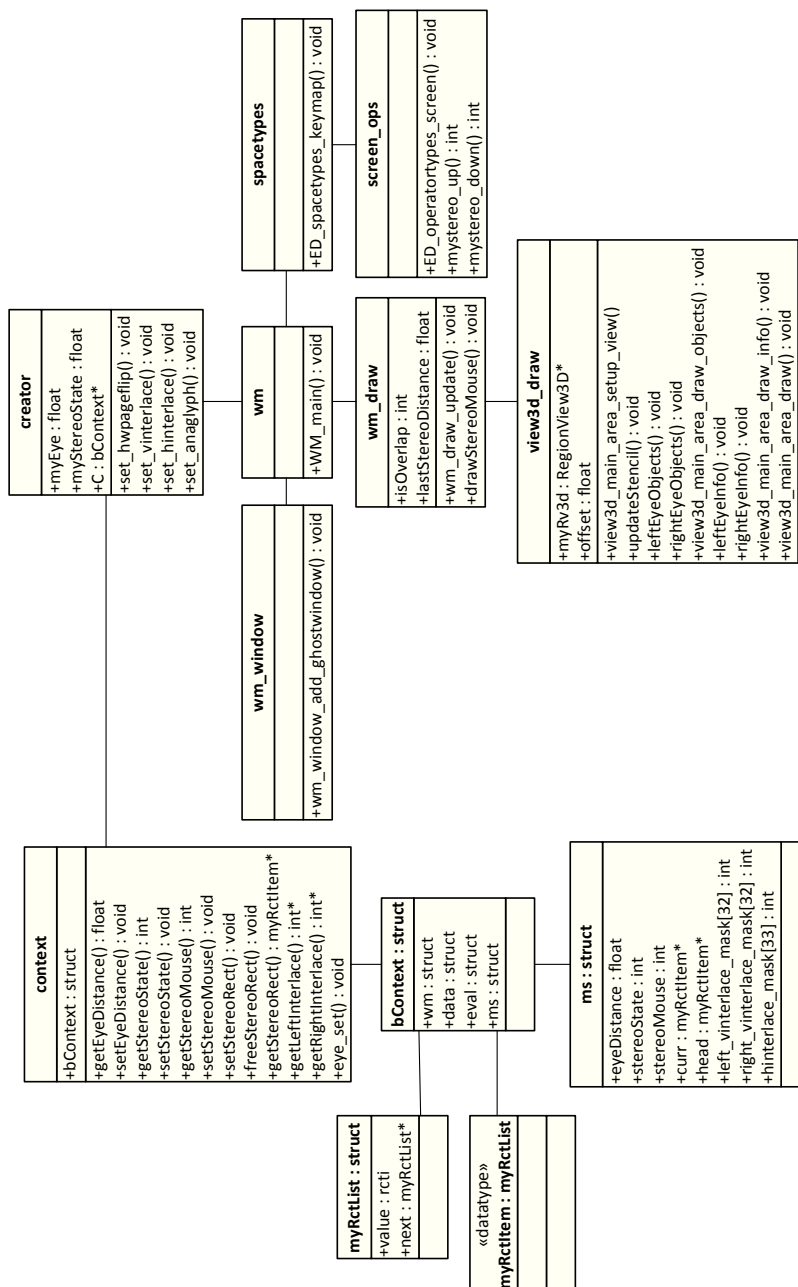
- [1] E. ANGEL. *Interactive Computer Graphics*. United States of America : Pearson Education, 2009. xxix, 828s s.
- [2] T. ANNEN, W. MATUSIK, H. PFISTER, H-P. SEIDEL a M. ZWICKER. *Distributed Rendering for Multiview Parallax Displays*. Saarbrücken, Cambridge.
- [3] 3D AUTO-STEREOSKOPICKÉ MONITORY. gali-3d.com [online], 2005-2011. Dostupné z WWW: <http://cs.gali-3d.com/autostereoskopie-3d/>
- [4] J. BLANCHARD, R. TSUNETO. *Stereoscopic viewing*. hitl.washington.edu [online]. Dostupné z WWW: <http://www.hitl.washington.edu/scivw/EVE/III.A.1.b.StereoscopicViewing.html>
- [5] C. BIANCO. *How Vision Works*. howstuffworks.com [online], 1998-2012. Dostupné z WWW: <http://science.howstuffworks.com/environmental/life/human-biology/eye10.htm>
- [6] BLENDER. *Blender*. blender.org. Dostupné z WWW: <http://www.blender.org>
- [7] BLENDER. *Building Blender*. blender.org. Dostupné z WWW: http://wiki.blender.org/index.php/Dev:Doc/Building_Blender
- [8] *Blender*. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Blender>
- [9] P. BOURKE. *Calculating Stereo Pairs*. paulbourke.net [online], 1999. Dostupné z WWW: <http://paulbourke.net/miscellaneous/stereographics/stereorender/>
- [10] P. BOURKE. *Stereographics theory*. paulbourke.net [online], 1999. Dostupné z WWW: <http://paulbourke.net/exhibition/vpac/theory.html>
- [11] P. BOURKE. *Stereoscopy, Theory and Practice*. paulbourke.net [online], 2007. Dostupné z WWW: <http://paulbourke.net/papers/vsmm2007/>
- [12] M. BRAIN. *How 3-D Glasses Work*. howstuffworks.com [online], 1998-2012. Dostupné z WWW: <http://www.howstuffworks.com/3-d-glasses.htm>
- [13] Ch. BUNGERT. *Stereoscopic 3D Virtual Reality Homepage*. stereo3d.com [online], 1998. Dostupné z WWW: <http://www.stereo3d.com/3dhome.htm>
- [14] S. DAVIS. *E2 3-D*. sue-davis.net [online], 1998. Dostupné z WWW: <http://www.sue-davis.net/e23d/index.html>
- [15] *Gali-3D*. gali-3d.com [online], 2005-2011. Dostupné z WWW: <http://cs.gali-3d.com>
- [16] GHOST. letworyinteractive.com [online]. Dostupné z WWW: <http://www.letworyinteractive.com/blendercode/d5/d2e/GHOSTPage.html>

-
- [17] M. HALLE. *Autostereoscopic displays and computer graphics*. Massachusetts, 1997.
- [18] N. HOLLIMAN. *3D Display Systems*. University of Durham, 2005.
- [19] A. D. KARTCH. *EFFICIENT RENDERING AND COMPRESSION FOR FULL-PARALLAX COMPUTER-GENERATED HOLOGRAPHIC STEREOGRAMS*. Cornell University, 2000.
- [20] Kognitivní server. fim.uhk.cz [online]. Dostupné z WWW: <http://fim.uhk.cz/cogn/?Module=dictionary&Letter=B>
- [21] L. LIPTON. *StereoGraphics Developers Handbook*. Stereographics Corporation, 1997.
- [22] Python. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Python>
- [23] P. RIESS. *Stereo Basics*. instantreality.org [online], 2007. Dostupné z WWW: <http://doc.instantreality.org/tutorial/stereo-basics/>
- [24] S. SCHNEIDER. *Stereoscopic Rendering in Blender 2.6*. noeol.de [online], 2011. Dostupné z WWW: <http://www.noeol.de/s3d/>
- [25] D. SINCLAIR. *GHOST Intro*. blender.org [online]. Dostupné z WWW: http://projects.blender.org/docman/view.php/13/4/ghost_doc.html
- [26] V. SKALA, J. STEFANY a V. BYSTRICKÝ. *Blender 3D Modelling - stereoscopic version*. Západočeská univerzita v Plzni, 2010. Dostupné z WWW: <http://www.kiv.zcu.cz/cz/vyzkum/software/>
- [27] E. SOJKA. *Počítačová grafika II*. mrl.cz [online]. Dostupné z WWW: <http://mrl.cz/people/sojka/pocitacova-grafikaII.pdf>
- [28] *Stereofotografie - třetí rozměr ve fotografii*. fotokamzici.com [online], 2008. Dostupné z WWW: <http://www.fotokamzici.com/aktuality/stereofotografie-treti-rozmer-ve-fotografii/222>
- [29] *Stereoscopic Parallax*. 3d-forums.com [online], 2009. Dostupné z WWW: <http://www.3d-forums.com/threads/stereoscopic-parallax.4/>
- [30] *Stereo Tutorial*. captain3d.com [online]. Dostupné z WWW: <http://www.captain3d.com/stereo/html/tutorial.html>
- [31] *Stereovision*. axon.physik.uni-bremen.de [online], 1994-2004. Dostupné z WWW: <http://axon.physik.uni-bremen.de/research/stereo/index.html>
- [32] P. TIŠNOVSKÝ. *Grafická knihovna OpenGL (13): double-buffering*. root.cz [online], 2003. Dostupné z WWW: <http://www.root.cz/clanky/opengl-13-double-buffering/>

- [33] UPPSALA UNIVERSITET. *Optics of the eye*. Uppsala University, 2004. Dostupné z WWW: <http://www.neuro.uu.se/fysiologi/gu/nbb/lectures/EyeOptics.html>
- [34] J. WATTIE. *Theoretical Stereo camera rig*. nzphoto.tripod.com [online], 1999. Dostupné z WWW: <http://nzphoto.tripod.com/3d/201camerarigs.html>
- [35] D. WEXLER. *Copy of Stereo Rendering*. nvidia.com [online], 2006. Dostupné z WWW: http://www.nvidia.com/object/IO_36545.html
- [36] *Z-buffering*. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001. Dostupné z WWW: <http://en.wikipedia.org/wiki/Z-buffering>

A UML struktura

Tento diagram popisuje provázanost jednotlivých částí programu Blender, se kterými se v průběhu implementace pracovalo. Jsou zde znázorněny všechny proměnné, struktury a procedury, které byly zmiňovány v kapitole věnované implementaci. V diagramu jsou také zaneseny všechny nově vzniklé části, které byly v průběhu této práce vytvořeny.



B Dotazník

Finální verze dotazníku, který byl dáván k vyplnění testovaným uživatelům.

Jméno a Příjmení:		Způsob zobrazení: Anaglyph / Quad-Buffer
Věk:	E-mail:	Pohlaví: Muž / Žena

Tento dotazník vznikl za účelem získání zpětné vazby od uživatelů programu Blender a jeho programové nástavby umožňující stereoprojekci při modelování. Dotazník i programová nástavba jsou součástí diplomové práce s názvem „Možnosti 3D stereo projekce v Blenderu“.

1. Pracoval/a jste někdy s programem Blender?
 - a) Ano
 - b) Ne
2. Jaké je zaměření Vašeho studijního oboru?

a) Technické zaměření	b) Netechnické zaměření
Název oboru:	Název oboru:
3. Na jaké úrovni jsou Vaše dovednosti v 3D tvorbě a modelování (v jakékoliv aplikaci)?
 - a) žádné
 - b) základní
 - c) průměrné
 - d) pokročilé
 - e) velké
4. Zaznamenal/a jste rozdíl ve vnímání hloubky se zapnutou stereoprojekcí?
 - a) Ne
 - b) Mírný
 - c) Ano
5. V jaké vzdálenosti byly přibližně Vaše oči od monitoru / projekční plochy?
 - a) 0.5m
 - b) 1m
 - c) 1.5m
 - d) 2m
 - e) 2.5m
 - f) 3m
 - g) 3.5m
 - h) 4m
6. Při jak intenzivním osvětlení jste s touto stereoprojekcí pracoval/a?
 - a) tma
 - b) šero
 - c) průměr
 - d) dobré
 - e) jasné světlo
7. Na stupnici ohodnoťte, jak se Vám pracovalo v Blenderu s vypnutou stereoprojekcí.
 - a) velmi dobře
 - b) spíše dobře
 - c) beze změny
 - d) spíše špatně
 - e) velmi špatně
 Důvod :
8. Na stupnici ohodnoťte, jak se Vám pracovalo v Blenderu se zapnutou stereoprojekcí.
 - a) velmi dobře
 - b) spíše dobře
 - c) beze změny
 - d) spíše špatně
 - e) velmi špatně
 Důvod :
9. Setkal/a jste se už někdy se stereoprojekcí?
 - a) Ne
 - b) Ano, uveďte kde:

a. Kino	c. 3D televize
b. Počítač	d. Jinde:
10. Pokud jste se již se stereoprojekcí setkali, jak na vás působila? (např.: „nevnímám jsem ji“ nebo „bolely mě oči“ atd.)

.....
11. Jaké projekční zařízení jste používali pro zobrazování?

a) Monitor – typ:	úhlopříčka: rozlišení obrazovky:
b) Projektor – typ:	rozlišení obrazovky:
12. Jaká byla rozteč zobrazovaných obrazů, která Vám nejvíce vyhovovala? (číselný parametr při spuštění)

.....

C Výsledky dotazníku

ID	Metoda	Věk	Pohlaví	1	2	3	4	5	6	7	8	9	11	12	Rozdíl
U22	Quad	24	Muž	a	a	b	b	g	b	b	d	b	b	0,2	-2
U23	Quad	24	Muž	a	a	c	c	g	b	b	b	b	b	0,3	0
U24	Quad	22	Muž	a	a	c	c	g	b	c	d	b	b	0,25	-1
U25	Quad	22	Muž	a	a	c	c	g	b	a	d	b	b	0,4	-3
U26	Quad	22	Muž	a	a	b	b	g	b	b	d	b	b	0,15	-2
U27	Quad	22	Muž	a	a	b	c	g	b	b	b	b	b	-0,25	0
U28	Quad	22	Muž	a	a	c	c	g	b	a	a	b	b	0,2	0
U29	Quad	21	Muž	a	a	c	c	g	b	c	d	a	b	0,25	-1
U30	Quad	21	Muž	a	a	c	c	g	b	a	b	b	b	0,2	-1
U31	Quad	21	Muž	a	a	c	a	g	c	c	c	a	b	0,25	0
U32	Quad	21	Muž	a	a	d	b	g	c	c	a	b	b	0,25	+2
U33	Quad	20	Muž	a	a	c	b	g	d	c	d	b	b	0,25	-1
														Avg.:	-0,75

ID	Metoda	Věk	Pohlaví	1	2	3	4	5	6	7	8	9	11	12	Rozdíl
U34	Quad	19	Muž	a	a	b	c	b	d	b	a	a	a	0,3	+1
U35	Quad	18	Žena	a	a	b	c	a	d	d	d	b	a	0,2	0
U36	Quad	19	Muž	a	a	e	b	a	d	a	b	b	a	0,2	-1
U37	Quad	22	Muž	b	a	c	c	a	d	c	b	b	a	0,25	+1
U38	Quad	23	Muž	b	a	a	c	a	d	c	b	b	a	0,2	+1
U39	Quad	23	Muž	a	a	d	c	a	d	c	a	b	a	0,2	+2
U40	Quad	26	Muž	b	a	c	c	a	d	d	a	b	a	0,3	+3
U41	Quad	27	Žena	b	a	a	c	c	c	c	c	b	a	0,3	0
U42	Quad	26	Muž	a	a	b	b	a	c	b	d	b	a	0,33	-2
U43	Quad	29	Muž	a	a	b	c	a	b	a	a	b	a	0,2	0
U44	Quad	22	Muž	b	a	a	c	c	b	a	a	b	a	0,2	0
U45	Quad	25	Žena	a	a	a	c	a	b	d	b	b	a	0,2	+2
U46	Quad	23	Muž	a	a	b	c	a	b	b	b	a	a	0,25	0
														Avg.:	0,53